

QL Today

Volume 12
Issue 2
December 2007
- Feb. 2008

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...

**THE QL
TEAM
OUR
MERRY
AND
BEST
2008!**

**TODAY
WISHES
READERS
XMAS
ALL THE
FOR**



www.QLToday.com

Contents

3	Editorial	
4	News	
10	End of the Season	Geoff Wicks
11	Hit me with your QL Stick?	Wolfgang Lenerz
14	Mapping Ancient and Modern - Part 1	Geoff Wicks
16	Letter-Box	
18	QemuLator In QoLour	Dilwyn Jones
20	Executable Things - Part 2	George Gwilt
31	Negotiation, Negotiation, Negotiation	Geoff Wicks
37	Off Topic - Version Control	Norman Dunbar
42	Minimalist Games - Part 2: Horses	Stephen Poole
42	My Holiday Romance	David Denham
52	Curve Recognition	Stephen Poole
56	Byts of Wood	Roy Wood

QL Today

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software Tel. +49 203 502011
 Kaiser-Wilhelm-Str. 302 Fax +49 203 502012
 47169 Duisburg email: JMerz@j-m-s.com
 Germany email: QLToday@j-m-s.com

English office:

Q Branch Tel. +44 1273 386030
 20 Locks Hill Mobile +44 7836 745501
 Portslade Fax +44 1273 381577
 BN41 2LB email: qbranch@qbranch.demon.co.uk
 United Kingdom email: QLToday@j-m-s.com

Editor:

Geoff Wicks Tel. +44 1332 271366
 5b Wordsworth Avenue email: gwicks@beeb.net
 Sinfyn email: QLToday@j-m-s.com
 Derby DE24 9HQ
 United Kingdom

Co-Editor:

Bruce Nicholls Tel +44 20 71930539
 38 Derham Gardens Fax +44 870 0568755
 Upminster email: qltoday@q-v-d.demon.co.uk
 Essex RM14 3HA email: QLToday@j-m-s.com
 United Kingdom

QL Today is published quarterly, our volume starts middle of September. Please contact the German or English office for current subscription rates or visit our homepage www.QLTODAY.com.

We welcome your comments, suggestions and articles. YOU make **QL Today** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

QL Today reserves the right to publish or not publish any material submitted. Under no circumstances will **QL Today** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL Today**. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

This magazine and all material within is © copyright 2007, 2008 Jochen Merz Software unless otherwise stated. Written permission is required from the publisher before the reproduction and distribution of any/all material published herein. All copyrights and trademarks are hereby acknowledged.

If you need more information about the UNZIP program which is used by our BOOT program to unpack the files, we suggest that you visit Jonathan Hudsons web site where you find more information about lots of interesting QDOS software and INFOZIP at www.bigfoot.com/~jrudson/

**The deadline for the next issue is the
10th of February 2008**

Advertisers

in alphabetical order

Jochen Merz Software	9
QBranch	34, 35
Quanta	25
RWAP	41
TF Services	49

On my way to the Eindhoven show I stayed overnight in Utrecht. In my hotel room was a notice written in perfect English apart from one delightful sentence that had eluded the eagle eye of the proofreader:

"In the bar you can enjoy a selection of whines by the glass..."

There is something special about this issue of QL Today, although probably no reader will notice it. It is the first issue for some time that we have not had an apology or correction towards the end of our news pages.

The average issue of QL Today contains about 35,000 words and it is inevitable that these will include some typing errors, spelling mistakes, homophones or even factual inaccuracies. Surprisingly few errors are corrected by readers. Most we spot ourselves when it is too late to correct them.

We strive to maintain a high quality in QL Today and part of the quality control is proofreading. There was a minor crisis in the proofreading of the last issue. Our Co-editor, Bruce Nicholls, who does the proofreading, was away on an overseas business trip. In fact he took QL Today with him to proofread on the plane, but just in case anything went wrong Jochen asked me to proofread the issue as well.

I learnt a lot from this exercise. The main conclusion was that, as I expected, Bruce is a far better proofreader than I am.

Proofreading is not easy. The proofread copy came as three PDF files totalling almost 6Mb. Although this may seem a massive download, it is only a fraction of the size QL Today would be if produced at the highest resolution. To keep the bandwidth down the resolution of the proofread copy was much lower and resembled the quality of a cheap dot matrix printer in draft mode. Illustrations were almost impossible to read.

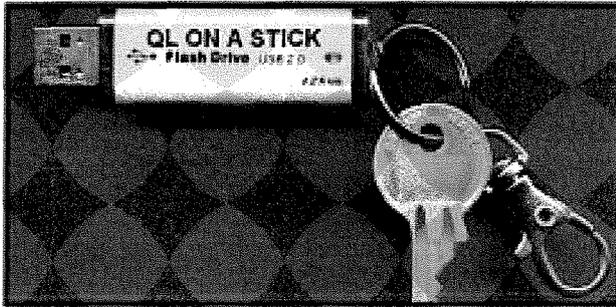
You cannot proofread at your normal reading speeds, but have to go through the text line by line looking for mistakes. Not an easy task when you are on your way to a foreign business trip.

When I took over the editorship of QL Today Jochen offered to send me the proofread copy of each issue, but I declined because at the time I was on pay as you go dial up. Now that I have broadband, I have asked to have the proofread copy as well as Bruce's corrections. In this way I can look for patterns of errors and hopefully can remove these at an earlier stage in the editorial and production process.

At QL Today we strive hard to bring you a magazine of high standard. We have said it many times before, but we are lucky to have a group of enthusiastic and loyal writers. Doing justice to their efforts is the task of the whole QL Today team. It is often the work the reader does not see that makes the difference between a good or bad quality product.

Key Ring QL

At the Birmingham Workshop **Dilwyn Jones** released "probably the most portable QL yet". This is a a USB memory stick containing three public domain QL emulators and software to "plug and play" in practically any PC.



The so called "QL on a Stick" was first proposed about 3 years ago by Dilwyn Jones and Darren Branagh and aroused much interest at the time, but never got off the ground. Dilwyn tells QL Today:

"A sticking (!) point at the time was the cost of USB pen drives. Darren Branagh and I were unsure if we could justify investing in sufficient pen drives to make this a viable project. It needed a minimum of a 128MB memory drive, which were expensive at the time. Since then, the price of these has gone into freefall (good job we didn't buy them then!) - you can now buy a 1GB memory drive for about £10 with indications that although the price may not come down much more, what may happen is greater capacity for similar cost in future."

Much of the early discussion centred on the cost and design of the drives rather than the software content. At one stage there was a proposal to approach Quanta for funding of the project but it proved impossible to draw up a suitable business plan because of a lack of people willing to help in developing the software.

Dilwyn continues:

"Accordingly, the project came to a halt. Then earlier this year the subject cropped up again on the mailing list and someone (I can't remember who) suggested the idea that the project could be made available on a CD-ROM and users could copy it to a pen drive of their choice.

So, I gathered together QLayer2 (which is a free QL emulator for Windows), QemuLator for Win-

dows (Shareware version) and the QPC2 demo version along with Marcel Kilgus's example QXL.WIN showing what QPC2's full commercial version can do - without spoiling it too much, I'll just say it's colourful and it even briefly talks to you. At the moment, I'm in discussion with Jimmy Montesinos, who updated QLayer to QLayer version 2 to see if I can use the more advanced QL2K derivative in future versions of QL On A Stick once development of QL2K reaches a stable enough state for general release.

QL On A Stick presents you with a simple menu to choose whether to start QLayer, QemuLator, or QPC2, or to read the QL manual (which is supplied as a PDF file and as plain text files), or to start a couple of useful QL-related programs running under Windows such as Jonathan Hudson's WXQT2 QL/PC file transfer program, and Norman Dunbar's Q-Stripper program which reads Quill DOC files and can convert them to plain text, HTML, PDF or DocBook XML files."

Dilwyn demonstrated the QL On A Stick to an interested audience at the Quanta Birmingham Workshop early in October. Present at the demonstration was Quanta chairman, John Mason, in case there was still a role for Quanta to play in the future development of the project. However this is now unlikely as the price of USB memory sticks has fallen sharply and it is easier to distribute the product as a CD whose contents can be copied to the disk. QL Today understands that one trader is examining the viability of producing the product on a memory stick as an alternative to a CD.

The development of the QL on a Stick was not always a simple matter. Dilwyn Jones explains:

"The preparation of QL On A Stick threw up a few interesting programming challenges. Probably the most interesting of which was handling the fact that when a USB memory drive is plugged into a PC, you have no idea in advance which drive letter the PC concerned will allocate to it. The prototype was drive K: on my desktop PC, but drive E: on my wife's laptop. The answer was to use relative path names, so QPC2 should not try to access C:\QXLWIN for example, but rather have its own QXLWIN in its own directory, so that instead of specifying drive names, I specify filename paths relative to where the emulator is started from.

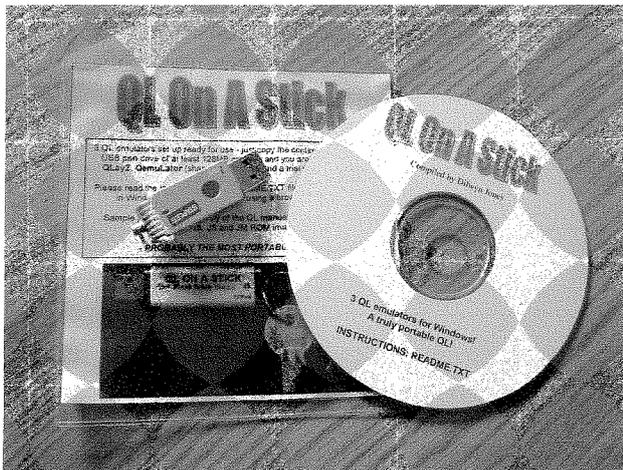
QLay2 and QPC2 make this quite easy - you just type them into the device assignment dialogues and away you go. QemuLator does not allow you to specify relative drive names directly, but it proved possible to manually edit its configuration files (the .QCF files) to specify relative paths and QemuLator accepts that.

The next step was to design a common boot program. This had to decide what was needed for each emulator. For example, QemuLator needs a specific mouse drive called Mac-Mouse11 to be loaded to use the PC mouse with pointer environment. I also wrote a little program launcher called Go! (it's actually code stolen from Launchpad's Quicklaunch menu system) as a user friendly front end for the emulators, and decided on a few essential extensions and toolkits to include, along with a few sample programs such as Turbo compiler, File Info 2, Xchange and a text editor/viewer, but tried to design it such that users could use these to get used to the systems and add programs of their own choice later."

There is also a copyright problem for USA QL-ers:

"Since the project includes a copy of the QL Manual, North American users need to apply for permission from Frank Davies and Paul Holmgren to obtain QL On A Stick, since they hold the rights to the QL manual in North America."

More information about the key ring QL can be found on Dilwyn's website:



www.dilwyn.uk6.net/qos/index.html

QL On A Stick is now available on a CD ready to copy to a USB pen drive of your choice for just 5 pounds either direct from Dilwyn Jones, or from Q-Celt Computing in Ireland.

Wolfgang Lernerz has reviewed this product for this issue of QL Today.

More Dilwyn News

Vision Mixer 2 Released

Dilwyn Jones writes:

"Vision Mixer 2 is now available from my website.

This is an updated version of my original Vision Mixer program of about 20 years ago. This release is for high colour modes (mode 32 and mode 33, and possibly mode 16, untested as yet on Aurora) systems only. The purpose is the same - screen picture slideshows with fancy visual effects for transitions from screen to screen. It can be used for presentations, slide shows, advertising display and so on. I use it, for example, to present a rolling slide show of information screens when I attend QL shows.

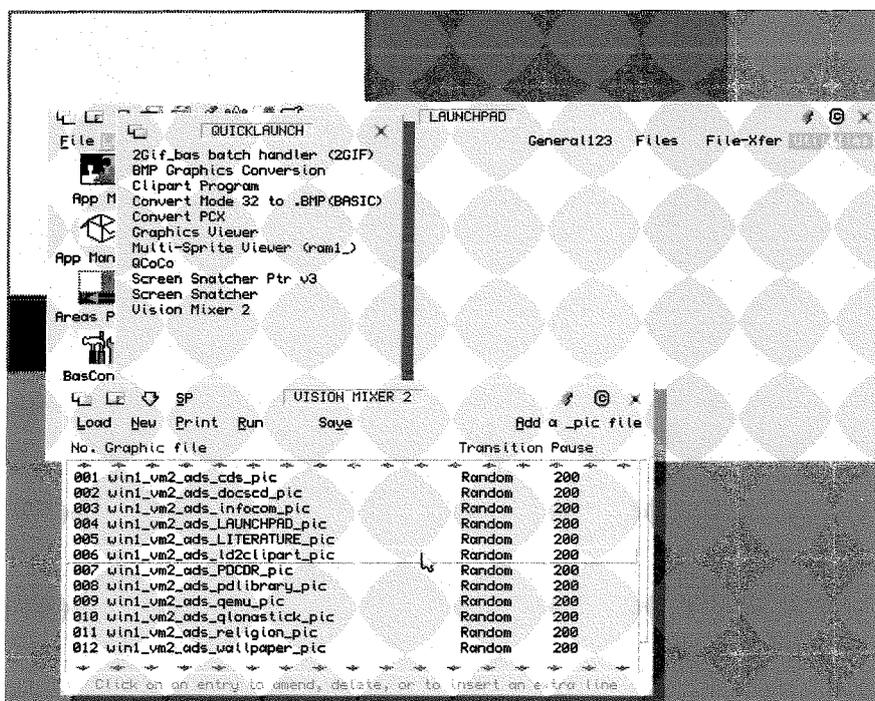
The program is pointer driven and needs high colour modes, Window Manager 2, plenty of expanded memory and Toolkit 2. All other extensions required (DJToolkit by Norman Dunbar, Display_Code extensions, QLiberator extensions, Easyptr 4 extensions and Basconfig extensions are all linked into the QLiberator compiled program VM2_obj) so you need not worry about any of those.

The user interface has been completely rewritten using Easyptr 4 to the extent that from the user's point of view this can be thought of as an original program, not an update, although the underlying concepts are broadly the same as the original Vision Mixer program from all those many years ago.

The fundamental improvements are:

- All new pointer driven, simplified user interface.
- Next screens always loaded from disk in real time during the viewing pause of the previous screens, frees up memory by not needing to hold all screens in memory, thus also allowing more pictures to be used in lists. Modern hard disk systems are pretty fast, so loading files in this way should not introduce much delay.
- Mixed screen mode screens may be used (mode 4, 8, 16, 32 or 33) with conversion done in real time at loading time, thus mode 4 screens may be used, with real time conversion to mode 32 done just before display using fast machine code routines from my BMP program.
- PIC files used, so that the program can determine picture size and mode number while loading, and so use just about any reasonable screen size.

- Not limited to 512x256 screen
- Can run in all screen resolutions - no direct writing to screen any more (uses pointer interface routines)
- Does not have to use the whole screen - VM2 can use just part of a high resolution screen, allowing you to work on the rest of the screen and Vision Mixer can keep running.
- If your system allows background screen redrawing (e.g. recent QPC2), VM2 can continue to run if part obscured without destructive screen overwriting (as Vision Mixer 1 would do).
- Screen names list and visual effects list all combined into one control file for convenience.
- Preview window for video effects selected.
- Can use System Palette colour themes.



Vision Mixer 2 is available from the Graphics page on my website, a 66K download from: www.dilwyn.uk6.net/graphics/index.html

PCB CAD Update

Version 6.26 of Malcolm Lear's PCB Cad program is now available to download. Malcolm describes this as a "major update." - Ground / power planes and complex component support. There are many internal hidden changes made to make a future PE version possible. PCB Cad 6.26 is a 586KB download from: www.dilwyn.uk6.net/graphics/index.html

EDITOR 2000 Correction

Mark Knight has informed Dilwyn that due to an error some files in the Editor 2000 package were from an older version. Mark has kindly made available a corrected release 3.07K of the rewritten Editor program, which is now available to download from his website at:

<http://www.the-furnace.talktalk.net/Contents/QL.html>
or from:

<http://www.dilwyn.uk6.net/editview/index.html>

In addition to the main program, the source files are also available for those who wish to study the code.

DJ Program Updates

Dilwyn's website now includes a number of program updates and some new programs.

ZIP MANAGER version 1.04 is now available for download from the archivers page on the website. This fixes a problem when running on level 1 filing systems where an attempt to extract the current directory name only worked on level 2 filing systems, and provides additional stack space to allow zipping/unzipping of larger numbers of files in one go.

Q-TRANS v2.09 is available for download from the Launchpad pages on the website. This also addresses a problem when running on level 1 filing systems, and also fixes a minor bug in the RENAME command.

LAUNCHPAD v2.09 has been released, which includes a revised way of logging in,

when using multiple user accounts, at the suggestion of Fabrizio Diversi. Registered users can upgrade by downloading the Launchpad package from the Launchpad page on the website, using the password previously supplied (password requests by email please if you are a registered user and wish to obtain a password to download the upgrade.)

GO! is a new program launching front end which is meant to be a pointer driven program launcher which is both easier to use than Launchpad and free. It lets you create a series of menus and sub-menus to automate the task of starting programs. It is File Info 2 aware, so assuming File Info 2 is suitably set up, you can even execute text or

graphics files! The Go! program is used as the front end in the QL On A Stick project.

The **QL ROMs page** now includes a Danish QL ROM image. It was meant to be there earlier this year, but somehow got accidentally deleted. Anyway, it's definitely there now, with grateful thanks to Klaus Frank who spotted the error initially.

The **BASIC programs page** now includes a handy little proportional spacing screen printing routine written in SuperBASIC, which lets you print in proportional spacing using QL fonts, rather than the fixed print pitch normally offered by the QL.

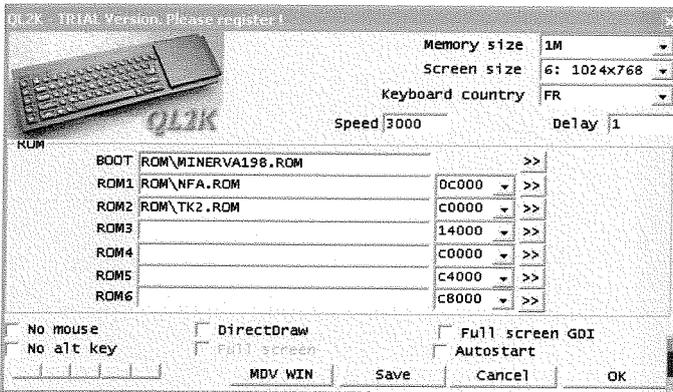
And finally Dilwyn has now pretty much completed the migration of the documents web pages from the original home on TopCities web pages to the main website, hosted by Freeola who have proved to be an excellent ISP so far. They now offer "unlimited" web space and Dilwyn is happy to recommend them to anyone wishing to set up a QL website. His site now consists of some 175MB of mostly zipped data and just seems to grow all the time.

Find his QL website at www.dilwyn.uk6.net

QL2K Upgrade

Jimmy Montesinos has posted a new patch for QL2K on his website

<http://www.jadlam.org/QL>



The details are:

Current version: 0.1 Build 0.98a (Alpha).

Changes:

- Adding a new debugger accessible by the menu (Command/Debug)... This debugger is self-documented and fits in a resizeable window.
- Completely rewritten execution timing (Now a QL set at original speed use very few percent CPU time in both DirectX or GDI mode). It is possible to set speed factor too from original QL Speed. To set speed factor use -P x switch where x is 1 for QL original speed, 2 for 2 time quicker and so..

- A new switch has been added to set the delay of MDV from 31ms to 1ms. This switch is -Q.
- DirectDraw FullScreen mode doesn't show window border anymore. We have left just the menu bar as often the ALT Key is locked by the emulator for QL usage.
- Many and many bug fixes and code cleaning.

SQLUG BUG

For technical reasons the SQLUG website, which hosts George Gwilt's software, temporarily cannot be updated. George Gwilt writes:

"New Version of GWASS

GWASS is now up to version 4.32. This new version was produced to avoid possible machine crashes when the code being assembled contains an error. The crash can occur when an error appears near the start of the code.

Normally the new version would be available on the SQLUG site, but this is not currently capable of accepting new items. GWASS v4.32 should be available instead on Dilwyn's site. The source files are in *gwassp15.zip* and the program files are in *gwasss15.zip*."

<http://www.dilwyn.uk6.net/asm/index.html>

Urs König Move

Urs König writes:

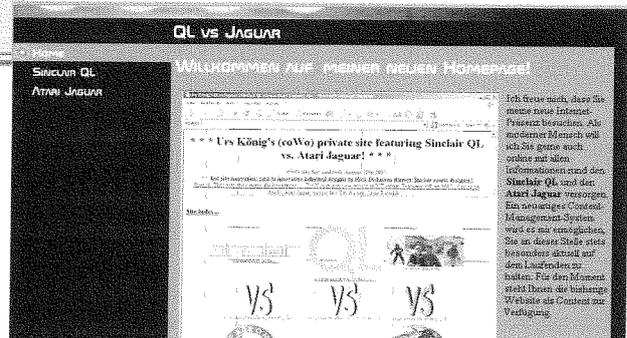
"Recently my ISP shut down the webserver which was the host for my website.

In the meantime I found another place to host my site.

My personal website "QLvsJaguar" can now be reached here:

<http://www.qivsaguar.homepage.bluewin.ch/>

Users please update your bookmarks, webmaster please update your link-pages."

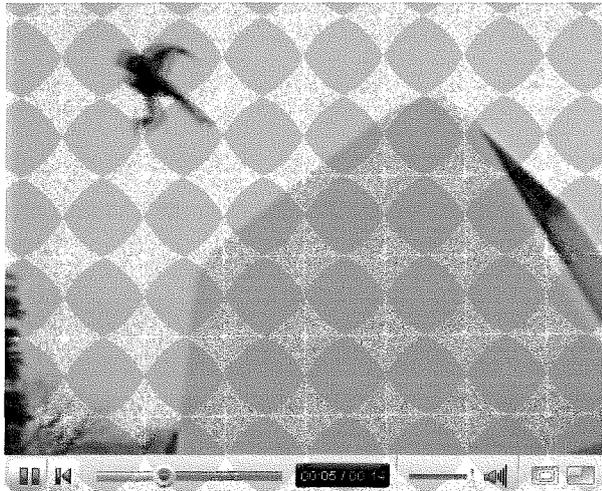


YOUTUBE Debut

The QL has made its YouTube debut. In the middle of August **Urs König** announced he had opened a "Sinclair QL" YouTube group.

Initially it just had two clips, the original QL advertisement of Clive Sinclair jumping over computers and a QDT demonstration by Jim Hunkins.

Sinclair QL TV advertisement 1984

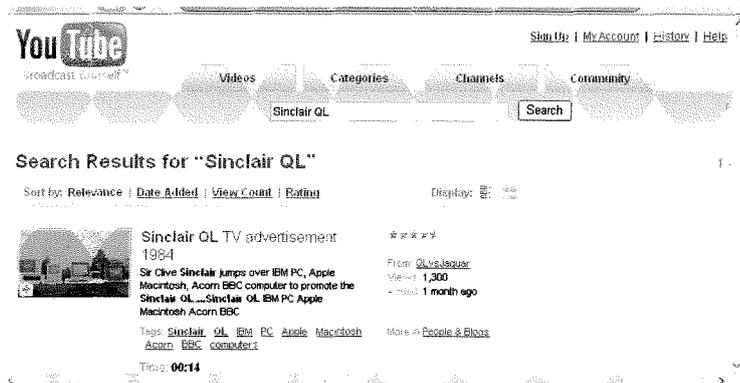


Within a few weeks it had grown considerably and now has Al Boehm's Midi demonstration at QL2000 as well as Simon Goodwin's Kodak camera presentation at the same show. There is also a Sinclair 1984 Rock and Roll clip. In addition Darren Branagh has placed the whole of the QL 2002 DVD in sections on the site.

QL2002DVD_02_The_Dealer_Room

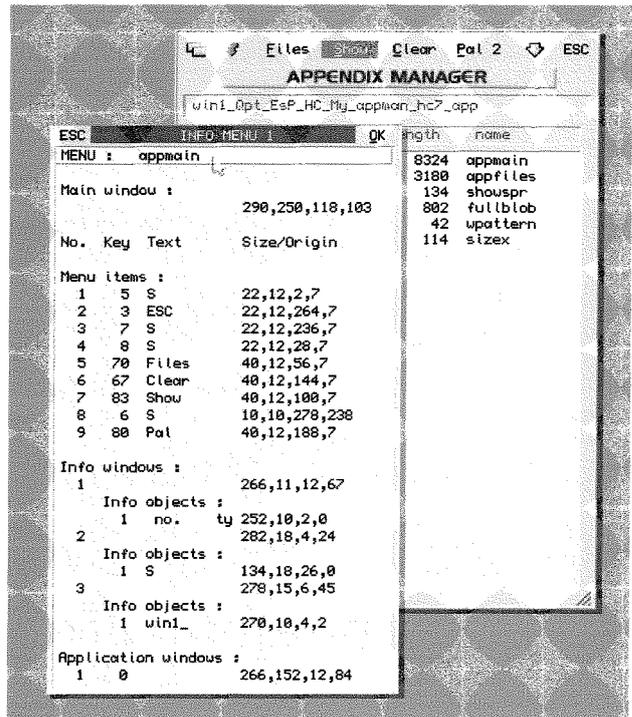


Go to www.youtube.com, click on groups and search for "Sinclair QL".



EASYPTR APPMAN Upgrade

Bob Spelten has upgraded the AppMan program in the EasyPTR suite. He writes:



"As you may know, the graphical interface for AppMan was modernised by Wolfgang Uhlig for use in the high colour modes. As an Aurora user I wanted it also to work in high resolution mode 4. So I changed it into version 4.06 with minor bug fixes and some added features to help all EasyPTR programmers.

For mode 4 the fancy title icons are replaced with text titles and a mode 4 resize icon was added. The background is set to black if the configured colour is not valid for this mode. The Main window will start at 1/16th from the top, centred horizontally and 400 pixels high, or 250 on a small screen. Resizing the Main window now also works correctly if it is selected with the CTRL-F3 key.

Like the 'Mode' button usage in high colour EasyMenu, a 'Pal' button is added to cycle through the system palettes 0 to 3 plus the inbuilt theme 4. So if you designed your menus with the \$2xx colour range, using the Show option will let you see them in any of these palettes, but only 0/3 for mode 4.

When showing a menu with a DO on the item number, this produces not the menu itself but a list with information about the menu; sizes of the windows, loose items, objects etc. When trying to show an item that is too big for the current screen this

Jochen Merz Software

Kaiser-Wilh.-Str. 302 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://SMSQ.J-M-S.com SMSQ@J-M-S.com

XMAS SPECIAL AT J-M-S! THE % SHOWN NEXT TO THE PRICE WILL BE GIVEN FOR ALL ORDERS UNTIL 31ST OF DEC. 2007!
E.G. QPC2 COSTS ONLY EUR 47,92 OR QPCPRINT ONLY EUR 35,91 INSTEAD OF EUR 39,90!

QPC2 Version 3 + SMSQ/E Software QL-Emulator for PC's	EUR 59,90	-20%
QPC2 Version 3 - Upgrade from QPC2 Version 2	EUR 19,90	-10%
QPC2 Version 3 - Upgrade from QPC2 Version 1	EUR 39,90	-15%
SMSQ/E ATARI or (Super)GoldCard or QXL	EUR 39,90	-20%
QPC Print - printer emulation driver for QPC	EUR 39,90	-10%
Agenda Agenda program for WMAN and Prowess	[V1.09] EUR 19,90	-25%
Suqcess Database front-end for WMAN	[V2.05] EUR 29,90	-25%
QD2003 Pointer-Environment-Editor	[VB.01] EUR 59,90	-35%
QD2003 Upgrade from QD98	[VB.01] EUR 10,00	
QD2003 Upgrade from previous versions	[VB.01] EUR 29,90	-20%
QMAKE Pointer-driven MAKE for GST/Quanta Assembler	[V4.31] EUR 19,90	-20%
BASIC Linker	[V1.21] EUR 19,90	-25%
WINED Floppy/Harddisk Sector- & File-Editor	[V1.26] EUR 19,90	-25%
FiFi II File-Finder - Extremely useful!	[V4.31] EUR 19,90	-25%
FiFi II Upgrade from Fifi V1, 2 or 3	[V4.31] EUR 10,00	
EPROM Manager	[V3.02] EUR 19,90	-25%
QSpread2003 Spreadsheet Program	[V4.04] EUR 59,90	-35%
QSpread2003 Upgrade from QSpread2001	[V4.04] EUR 10,00	
QSpread2003 Upgrade from V1	[V4.04] EUR 39,90	-20%
QPAC I Utility programs	[V1.11] EUR 24,90	-25%
QPAC II Files, Jobs & other Things	[V1.45] EUR 39,90	-25%
QTYP II Spell checker	[V2.17] EUR 34,90	-25%
QPTR Pointer Toolkit	[V0.30] EUR 39,90	
DISA Interactive Disassembler	[V3.04] EUR 39,90	-25%
typeset-ESC/P2 text87 driver for all ESC/P2 printers (incl. Stylus)	EUR 29,90	
CueShell	[V2.14] EUR 39,90	-35%
CueShell for QPC	[V2.14] EUR 20,00	-15%
SER Mouse software mouse driver for serial mice	EUR 10,00	
EasyPTR Version 4	[V4] EUR 59,90	-15%
EasyPTR Version 4 - Upgrade from earlier versions	[V4] EUR 39,90	-10%
text87plus4patch - now for QPC, QXL, Q40, Q60, Aurora	EUR 10,90	
QDT - QL Desktop program	EUR 59,90	-10%

We accept VISA, MasterCard & Diners Club online and offline! Amex only by mail or fax, not email!
New payment methods for our customers: Money transfer to "local" account in many countries!

- Deutschland: Jochen Merz, Account 493 50 431, Postbank Essen, BLZ 360 100 43
- Österreich: Jochen Merz, Account 85055317, PSK Wien, BLZ 60000
- Switzerland: Jochen Merz, Account 60-690080-4, PostFinance, Clearing-Nr. 09000
- The Netherlands: Jochen Merz, Gironummer 3258439, Postbank NL Amsterdam
- and from all other countries in EUR with IBAN and BIC to account
 Jochen Merz, Deutsche Postbank AG, IBAN: DE21 3601 0043 0611 1004 37 / BIC: PBNKDEFF 360
- UK customers can pay in £ (convert EUR prices above to £ by multiplying with 0.73) to
 Jochen Merz, Account 83795395, Citibank UK, Sort code 30-00-45
 or send cheques in £ - no fee for UK sterling cheques!
- US customers can pay in US\$ (convert EUR prices above to US\$
 by multiplying with 1.4) - no fee for US cheques in US\$!

Cheques payable to Jochen Merz only!
 Price list valid until 31st of Dec. 2007

used to kill the AppMan job but will now report the error and continue and may still produce the info list when a DO was used. Other errors with opening menus may stop AppMan but will report the error in most cases.

When viewing sprites they are now centred in the viewer window. A DO on this window will cycle through the current available- and selected background and the 8 basic QL colours. To make an information option also for sprites, I decided not to reinvent the wheel but to use Per Witte's excellent sprite viewer 'Sprv'. This has options to show sprite statistics and show multiple- or extended sprites. I adapted his program to accept the sprite title plus address and current palette, to be used instead of the inbuilt

viewer. I called it 'SprvAPM' and it's part of the package. More details on the new CMD\$ structure is in the Readme file.

A new configuration item is added for the full filename for an external sprite viewer like SprvAPM.

The configured default directory for files is only a starting point, once accessed, the directory for each file type will be remembered separately.

When an _app file is loaded, the APPn value is shown in the Files menu.

AppMan 4.06 itself is compiled with an _app code block, so the latest 'PtrMenr_cde' file must be loaded residently, which means you must be a legal owner of EasyPTR version 4. Contact JMS or Q Branch for this upgrade."

End of the Season

by Geoff Wicks



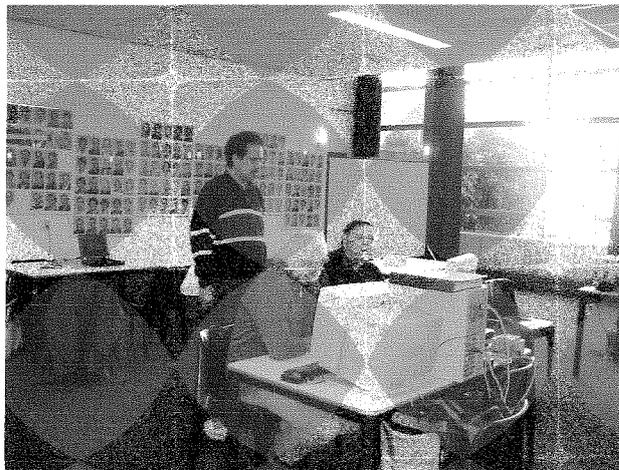
Life is full of changes, although fortunately some things remain constant.

The 2007 QL shows season ended at the end of October with the final Eindhoven show. This year my annual trip to Eindhoven brought many changes. Stena Line have restored the night ferry to the Netherlands so there was no more overnight sleeping on buses to save on hotel costs. I also changed the hotel I stayed in Utrecht.

There was a proposal for the October meeting to be an "international" one, but for the first time in years the UK traders decided not to go to Eindhoven. Jochen Merz attempts to attend all Eindhoven shows and was present as usual, although a dodgy car battery meant he arrived late.

There is a sense that all Eindhoven shows are international. 12 people were present at the October show and came from the Netherlands,

Belgium, Germany, France and the UK. Eindhoven is the great constant of the QL community. The St. Joris college has hosted more shows over a longer period and at a greater frequency than any other venue.



Eindhoven may not always attract large numbers, but a lot of work gets done there. People are to be seen in small groups discussing projects, exchanging ideas or just passing the time of day. Steve Poole, Jochen and I had some valuable conversations about QL Today.



Quite a few of the attendees had travelled long distances to Eindhoven and that led one voice to suggest that perhaps we should stop attending QL shows as a way of reducing our carbon footprints! (Now let's think positive. Would anyone like to write a program for QL Today to enable our readers to calculate their carbon footprint?)

After the show we discovered another Eindhoven tradition had changed for ever. The after-show Chinese-Indonesian buffet is no more. The restaurant has been taken over and now has Chinese-Suriname cuisine.

Just a final thought on Eindhoven. Quanta is planning a major event to celebrate the QL's quarter century in 2009. Given that few overseas QL-ers now attend Quanta shows, even major ones, would there be an interest in similar event for continental QL-ers?

One small change: Eindhoven Shows will start in 2008 at 11am instead of 10am ... to take account of battery problems, traffic jams etc.

Hit me with your QL Stick? Testing the QL that comes on a stick

by Wolfgang Lenerz

Having "zero footprint" software is becoming a major fad in the computer world. This is basically software that you don't need to install on your computer. You just plug a USB memory stick in, with the software pre-installed on it, and you execute it from there. The software will not install on the computer nor, normally, leave any trace of itself (hence the "zero footprint") unless you direct it to store some data. I'm not so sure whether this is really useful or not, but I presume that some use can be made of such nomad computing. Of course, in the QL world, it is rarely needed to install software as such, but if you use Windows or Linux, you'll know that this isn't like that there. Anyway, this fad has now also reached the QL world in a roundabout way - we now have a "project" that puts certain QL emulators on a USB stick, so that you can use them right from there. Since these emulators work under Windows this of course presumes that you use a Windows PC under which all of these emulators can then run. The stated purpose of this to "give you that ability to carry QDOS or SMSQ/E around with you wherever you go" - a laudable intention. I presume that one of the purposes of such a project is also to lure some non-QLers (back) into our fold, an even more laudable intention! I had a look at this USB-stick QL.

Installation

Installation was pretty simple. The software pack arrived as a huge zipped file. In most cases, you'll probably get the software on a CD where it is already unzipped, so all you have to do is copy it onto the USB drive. I just unzipped the file (under Windows) onto a blank 512 MB USB stick. As USB sticks go, 512 MB is (these days) considered to be pretty small, but up to now I haven't found any need to upgrade to anything larger. I'm sure that as soon as I write this, some software will come up that I'll need and that needs a larger pendrive, but I digress...

Once you have unzipped/copied everything, you will find a nice "readme.txt" file in the root directory of the USB stick, which I studiously ignored just like about everybody I know. The admitted way of doing things with regard to readme files seems to be to only read these files if something goes wrong. I just looked at the content of the USB drive, there was a batch file called "go.bat" on it and I just double clicked on that (I did run all of the files through an anti-virus check first - no virus reported). This then threw up a small menu, which you can see in the picture shown in fig. 1 and which I call the "initial menu screen". It propo-

ses a choice between three well established QL emulators: QPC, Qemulator and Qlay2. Please note that the first two of these are commercial products whereas Qlay2 is freeware. The initial menu screen also allows you to read the readme.txt file, read the scanned QL Users guide which is contained as a pdf file on the stick or use the Qstripper program (and no, that name promises far more that it keeps - it's used to strip Quill doc files of control codes that modern PC or windows programs can't use.).

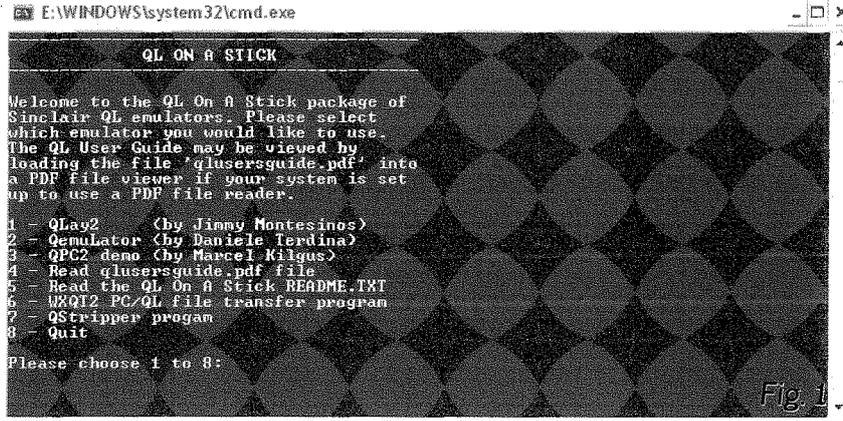


Fig. 1

Pushing buttons

Feeling adventurous, I just pushed one of the buttons. Being a long-time QPC user, I first of all hit the "3" key. This brought up the typical QPC initial configuration screen. There was no fuss, no "drive not found" or any other problem. The batch file called QPC. Clicking OK on the QPC configuration screen then brought up QPC running a demo, as you can see in screenshot 2. What you

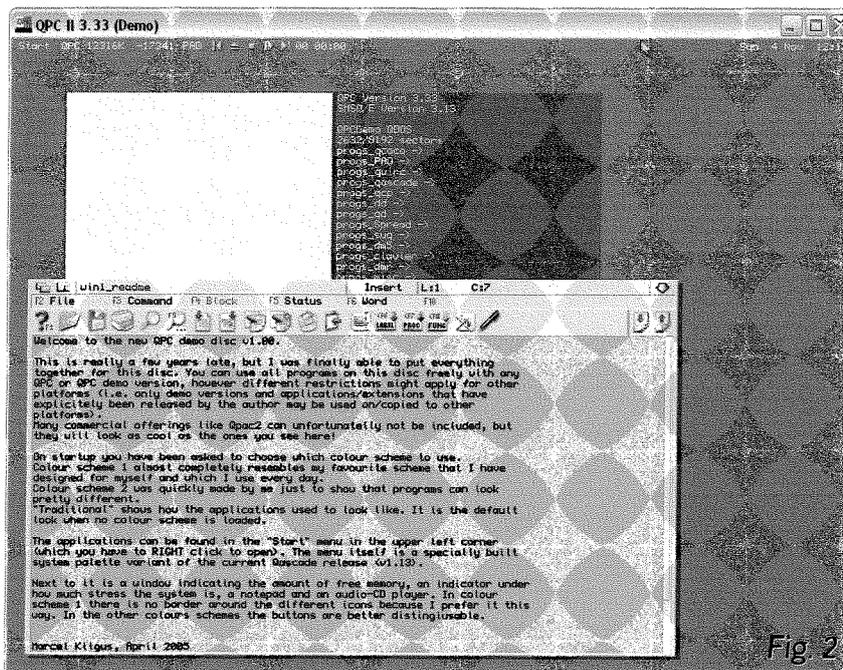


Fig. 2

can't see in that screenshot is the nice voice of a lady telling me to have fun with QPC 2, a voice actually played under QPC2 itself with the Sbasic command "SPL dev1_qpc2_ub, SOUND1". Nice. The QPC demo has a demo "hard disk" already set up and it booted from that without any fuss. This was a real one-click solution!

Since this here isn't a review of the different emulators, I won't go into many further details of them. QPC is the one I personally prefer. It runs SMSQ/E perfectly so it has the pointer environment already built in. It also, of course, has the high colour modes and high resolution screens, a sound system (which produced the sound mentioned above) and TCP/IP access. It can also read/write DOS/Windows floppies and hard disks, and can use the serial and parallel ports if your PC has them. However, being a demo version, this version of QPC2 supplied here can't write to disk, so you can use it to run programs for

testing/demo purposes, but you can't save anything. Apart from that it worked as expected - flawlessly. I do have a small niggle, though: When in the QPC demo, I typed in some things into the basic interpreter. I then noticed that my keyboard wasn't configured correctly - QPC was set up to expect a normal english keyboard, whereas I use a french one. So some of the letters came out wrong. Ok, this comment is probably a bit unfair, since QPC did bring up its configuration screen first of all, where you can set up the keyboard. However, if the purpose is to bring non QLers to use these emulators from the USB stick, this problem should somehow be addressed (I admit that I have no idea how) Anyway, I closed the QPC emulation after having satisfied myself that this worked alright. The first thing I then noticed was that the initial menu screen was gone. Apparently as soon as you choose one the the options it proposes, it calls up that option and then exits. Perhaps it would have been better to have it permanently there.

I recalled the initial menu screen by re-clicking on the go.bat file and tried to run Qemulator. This also came up nicely (see screenshot 3). It goes into a boot program that nicely sets it up, giving you the option to install several programs (eg the pointer environment). It worked flawlessly as well. Qemulator is a shareware emulator which can be either used in a free of charge but limited mode (speed restricted to that of an original QL with a maximum of 384K RAM, which is already OK for a basic QL) or, upon registration, as a fast QL emulator. It then has larger RAM support and is able to use QDOS (versions JM, JS and Minerva supplied, complete with Toolkit 2) or even SMSQ/E for the Gold Card (Aurora version needed for 256 colour mode support). The

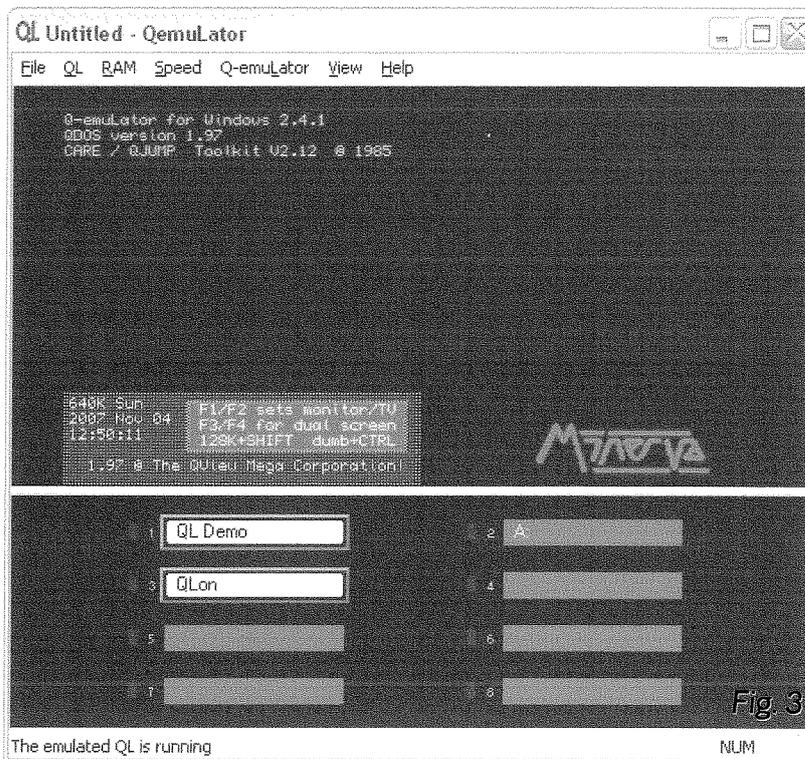


Fig. 3

registered version supports level 2 directory systems, can access QXLWIN or native file systems, has a builtin TCP/IP system and, like QPC, can use the serial and parallel ports if your PC has them. It didn't have the keyboard problems, all letters came out correctly.

After leaving Qemulator, I recalled the initial menu screen by re-clicking on the go.bat file and tried to run QLAY2. However, this came up with an error. I was unable to find what the problem was. A version of QLAY2 downloaded from the author's website didn't have this problem and QLAY2's initial configuration screen came up OK (see screenshot 4).

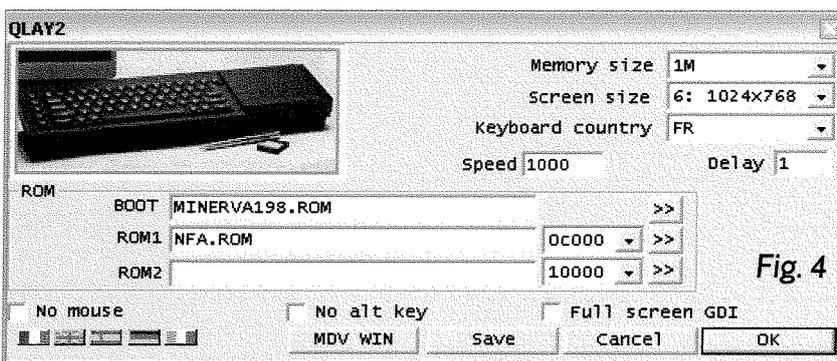


Fig. 4

Apart from allowing you to run the three emulators and reading the documentation, the initial menu screen also allows you to run Qstripper, mentioned above, and WXQT2, a QL/PC file transfer program.

How to get it?

The information provided at <http://www.dilwyn.uk6.net/qos/index.html> tells us that his project is published by Q-Celt Computing in Ireland and to contact Darren Branagh (send email to dbranagh@kingfisherclub.com)

for prices and any enquiries about supplying the software on CD-ROM or ready to go on a USB pen drive. It would probably make sense to get it directly on a USB stick, it seems, from discussions on the QL Users list, that some sticks can be provided pretty cheaply. Apart from that, seeing that the programs supplied on the stick are provided free of charge by their authors, the price shouldn't be very high.

Perhaps I just had a corrupted file in my zip file, it happens sometimes. Anyway, globally and this problem apart, the project of providing 3 QL emulators to run from a USB stick just works. It's a turnkey project any halfway competent Windows user (I know, I know, that's an oxymoron) can get to run.

On a final note, the true name of this project is "QL on a stick". Dilwyn Jones's website states that the name "QL On A Stick" is copyright © 2007 Darren Branagh and Dilwyn Jones so I've attempted to avoid infringing that copyright as much as possible in this short description thereof.

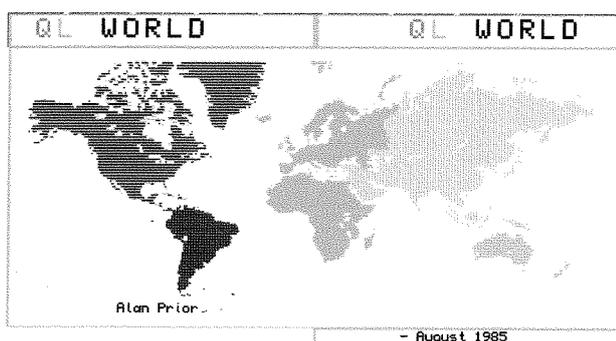
Mapping Ancient and Modern – Part 1

by Geoff Wicks

In next issue of QL Today we shall begin a major new series on mapping. It was originally our intention to start the series in this issue, but for technical reasons that I shall describe later, we decided to postpone it for one issue. Hugh Rooms is already well known to QL Today readers for his articles on GPS, and in this series he will write about Mercator Projection. The QL has always had programs for drawing maps, but Hugh has brought us to the brink of a major breakthrough in the potential of QL mapping software.

The history of QL mapping programs is not just about drawing maps. It is just as much about the creativity of QL-users and the programming flexibility that SuperBasic has given them. Quite often the QL has been in advance of the PC and, even when the QL could no longer rival PC graphics, SuperBasic programming has enabled QL-ers to be innovative in developing mapping techniques. This innovative spirit continues today in Hugh Room's pioneering work on GPS technology and map projection.

The first QL map drawing program dates from 1985, was written by Alan Prior and published in QL World. It used a simple technique to draw a map of the World. Tracing graph paper was used to copy a World map and the east and west coastlines of each country noted. These were then converted to co-ordinates on the QL screen and the LINE command used to draw the map.



We often forget that the QL had superior graphics to the PC when it was first launched. Even when PC graphics eventually caught up with the QL you needed deep pockets to be able to afford a machine capable of graphics of the same quality. When QL World published the program I was living in Amsterdam where once a year there was a major computer show increa-

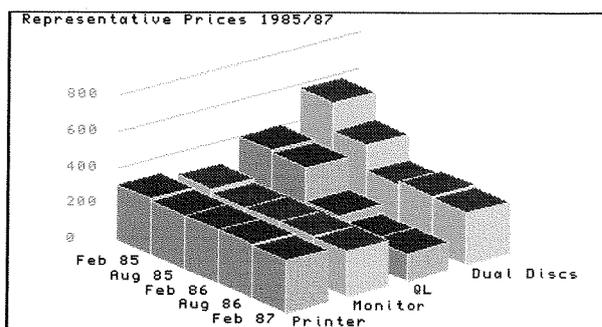
singly dominated by PCs. In a quiet corner you could see the QL among the "other" computers. The Dutch user group, Sin_QL_Air, used this program to demonstrate the QL's capabilities.

I used Alan Prior's technique in my General Election program. This has 641 lines of mapping data, one line for each electoral district. These can be printed in the colour of the winning political party to build up a political map of the whole country. It was a painstaking, time consuming technique, but once the data was present it was versatile. I could quickly program the data to show the areas of greatest strength for one of the political parties or even with GD2 colours the relative strength of the party in different areas of the country.



This mapping technique is simple, but it does have a serious shortcoming that Dilwyn Jones confronted me with earlier this year. But before I write about that, a slight diversion to look at 3d mapping.

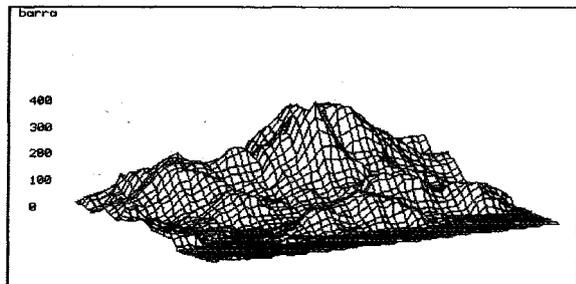
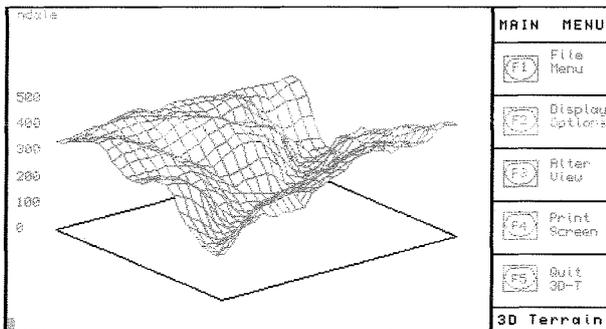
Again we are going to look at a program published in the early years of QL World, this time from 1987. Mike and Sue Lloyds' program displays 3D bar graphs that can be manipulated in various ways by changing the block width, the viewing angle and the viewing height.



I have a great affection for this program because shortly after it was published I used it to display the sickness pattern of various categories of worker when I was secretary of a works council. As we worked in a sector with a high sickness rate this was always a hot topic in the agency. We had a domineering director who was nervous of any attempt to raise the issue. On this occasion, heavily bitten by the price of PC programs, he decided to attack the messenger and not the message: **"And how much did this wonderful program cost?"**

It was a delight to see his face when I told him I had typed it in from a magazine. The QL had scored again!

But what is the relevance of this program to mapping? Once again we see creative QL thinking. A QL World reader, Ian Thompson, realised the program could be adapted to display contour lines. Together with Rich Mellor he wrote a commercial program "3D Terrain". This program is no longer on the market and it can be a little difficult to run on modern systems.

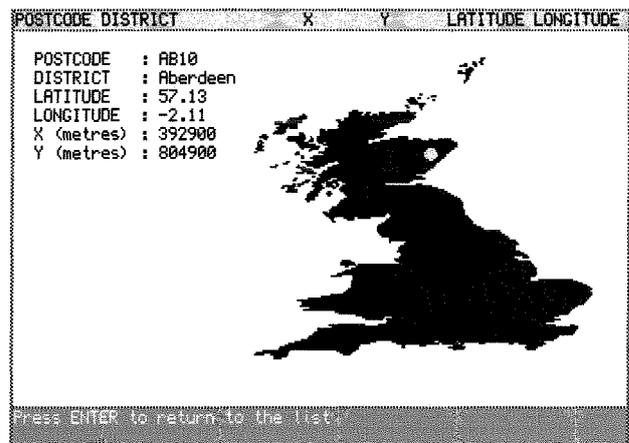


Just as with 2D maps collecting the data was a painstaking process using tracing graph paper or a ruler. You entered your contour data into abacus and the exported it in Easel format to import into the program. 3D Terrain came with the example of Nidderdale in North Yorkshire. As with the Lloyds' 3D program it was possible to alter the angle and height of view.

I used 3D Terrain to draw a 3D graphic of the island of Barra in the Outer Hebrides and sent it to one of the outdoor magazines. I suggested that if amateur programmers using a so-called obsolete computer could produce 3D maps then

Ordnance Survey could do the same. The letter was never published, but 13 years on there are now several companies, most notably Anquet, Memory Map and Fuwara, who sell mapping programs with a 3D option. Never forget the QL got in first!

But back to 2D mapping, and Dilwyn's challenge. Dilwyn had found a data base of postcodes on the internet, which among other things gave their latitude and longitude. He wrote a program using this data and then suggested I added a map based on my election map. This proved more difficult than I expected. Some towns were plotted accurately but others not. Portsmouth, a southern town, was spot on, but Aberdeen, a northern coastal town, appeared well inland.



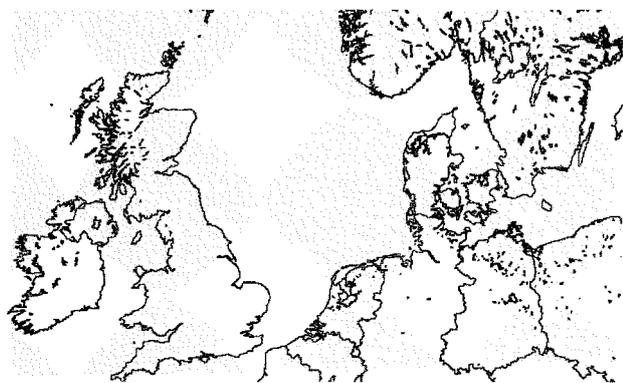
I knew my map was not 100% accurate as it had been taken from newspapers and not an atlas, but there was another factor at work. The problem occurred with latitude and not longitude. The more northerly a town, the more likely it was to be printed west of its actual location. Clearly the problem was the curvature of the earth. I added what I termed as a "bodge" to correct the x co-ordinate. This was done by trial and error and not scientifically.

Hugh Rooms has cracked this particular problem the scientific way and in so doing has opened up exciting possibilities in QL mapping programs. The mathematics are quite horrendous and Hugh had to relearn a lot of trigonometry. Do not be put off by this as Hugh has cleverly written an article that is of interest to the general reader, but which contains the detailed maths in appendices. It was the appendices that caught us out at QL Today. We felt that to do justice to Hugh's work, we would have to include all the maths, but this was not possible using our normal production techniques. The appendices contain numerous Greek letters and mathematical formulae that can get lost when transferred from one computer to another or even between programs on the same computer. We have spent many hours devising a

system that will enable us to reproduce these and we would like to thank Hugh Rooms for his patience and help when we were doing this. Even so laying out the article is going to be much more complicated than normal. It involves work for me in preparing the content in a suitable form for Jochen to lay out and then extra work for Jochen during the laying out process. We wanted to give ourselves the time to be able to do this with care.

But what can the reader expect from Hugh's work? One advantage is that you no longer have to compile the mapping data yourself, but can download it from the internet. However, the internet data files are huge and downloading is not for the fainthearted. I shall describe this process in the second part of this article.

The main advantage is maps of a much higher quality. As a taster we reproduce three maps of increasing magnification. They show the North Sea, part of Scandinavia and the Norwegian fjords.



We hope to have another benefit for our readers by delaying the publishing of Hugh's articles. It gives us more time to prepare cut down data



bases of the countries where QL Today readers live. Without giving away too many secrets, Just Words! is looking at the feasibility of adding a maps section to its website so that readers can try out Hugh's program for themselves.

Letter-Box

Mike Bedford-White writes:

My Thoughts on Workshops

My name is Mike Bedford-White. I hold the position of secretary and treasurer for the Birmingham subgroup and have been secretary for some 20 plus years. I also have been responsible for lecturers and workshops etc. for our meetings.

I am a virgin, (Titter Ye Not), with regard to QUANTA workshops, or was before the Birmingham do. The reason for this epistle is my thoughts on what can happen or not.

A little history:

I saw the request for a venue for the QUANTA AGM hopefully in the midlands in the Quanta Magazine. After mulling it over and asking one or

two pertinent questions, I offered the pub where the midland group meet, but this was just a bit late!

Some weeks later I had a phone call from the QUANTA secretary saying that normally a QUANTA workshop is held by the person or group to show that it can work at the possible venue.

After more discussion a date was set and preparations were put in hand. Unfortunately this did not come about although a workshop was hastily convened, in Solihull, to take its place.

Because of my "Virgin" status I did not have the necessary information or experience of the needs for a QUANTA AGM./workshop, although I did have some experience running various AGMs

and parties for the midlands group. Even though I had some conversations regarding the workshop I was still a bit in the dark. Because I had assumed that as it was an AGM there would be a maximum 35-40 people and the midlands group venue would have been ideal.

It was only when Geoff Wicks came to our venue that I learned what was entailed in the organisation and running of a workshop and the requirements thereof. Our venue was cosy but totally inadequate for a workshop.

When I knew of the needs I was able in less than 48 hours to sort out a suitable venue, although the date was not available due to the short lead time. It could have been had I fully known what was needed earlier.

My gripe is:

I am all in favour of the local groups running workshops but QUANTA needs to supply a pro forma for running a workshop (especially to virgins).

Groups who already run workshops on a regular basis, have no problem, as they already have the "LIST". Presumably these groups would or could put down on paper the arguments, reasons and experiences in running the workshops.

With the experience QUANTA has in running and helping to run workshops I would have thought it not too difficult to have to hand a logistical document, detailing what can be achieved, how it can be achieved and what resources are available from QUANTA i.e. funds, traders and demonstrators etc. What is required of the organiser in the way of input i.e. venue (size of room, power points overhead projectors), date, time (from - to), directions, refreshments, whether an on the door charge can or should be taken and a clear delineation of who asks traders, demonstrators etc. to take part.

It is no good keeping the information to yourself because if you do no-one else will be able to run or attempt to run a workshop or anything similar! Finally, there are probably not many workshop virgins out there but those that are need to be coaxed and warmed up for the event to take place, because if my experiences are anything to go by nobody else will volunteer!

George Gewilt writes:

Comments on Programming in Assembler - Part 19

In case Norman Dunbar expects me to comment on his latest enlightening and useful article I offer the following remarks.

1. I must admit to making all the same errors with the Bcc and CMP instructions that Norman says he makes. But I think I usually test D0.L correctly

after a TRAP or vector call. Incidentally, although testing D0.L is not necessary and is usually harmless after a call to a vector, doing so may cause problems. There are several vectors with no error return and in these cases D0.L may not be zero and the condition codes may not be set to signal zero.

For example the vector CN_FTOD converts a floating point number to ASCII. Pennel in his book *The Sinclair QDOS Companion* says

"Errors none - ignore value of D0 on return."

Also, the TRAP #1 call, MT_RCLCK, has no error but D0 can be non zero on return. So it is wise to eschew testing D0 whenever the routine has no error return.

2. Signed tests seem clear enough to me. Thus GE for "greater than or equal" is easy to grasp. Less easy to remember is CC for the unsigned version of GE. The same is true of CS for unsigned "less than". Here some assemblers (GWASS for example) accept LO as an alternative to CS. LO stands for low and is easier to remember.

3. The CMP instruction in Motorola's manual has the description:

"Subtract the source operand from the specified data register and set the condition codes according to the result."

Does this make it easy to interpret CMP D0,D1?

To interpret the instruction correctly you have to remember that here the source operand is D0 and the specified data register is D1 and not the other way round.

My brain hurts at least as much as Norman's at this point. The way I remember what CMP really does is first to remember that

```
CMPI.W #1234,D0
```

compares D0 with 1234, just as Norman says. But it seems obvious to me that to test whether D0 is less than 1234 I use the test BLT. I then force myself to believe that in `CMP.W D0,D1` it is the second operand which is being compared with the first. So to test if D1 is less than D0 I use BLT.

I actually think that the business of CMP subtracting one operand from the other is confusing. What does it mean that the result of the subtraction is "greater than". Greater than what?

4. The DBcc loops do not confuse me, perhaps because I interpret them as "branch until condition". Thus DBEQ means branch until equal. Also DBF means "branch until false" which can never happen so that the branch occurs the full number of times.

QemuLator In QoLour

by Dilwyn Jones

Recent versions of Daniele Terdina's QemuLator QL emulator for Windows have advertised the possibilities of higher colour depths than the standard QL mode 4 and mode 8. In this article I hope to show how to make use of this facility.

The current v2.4.1 release of the emulator hints at additional video modes, offering the possibilities of Aurora and Q60 video hardware emulation, plus the ability to run SMSQ/E. In practice, QemuLator cannot run a Q60 version of SMSQ/E at the moment, so the Q60 mode is in practice restricted to writing direct to the Q60 compatibility areas and it ends there - no 16 bit colour unless you write your own routines for direct access of the video memory areas. I think that the facility was only added for a private project being undertaken by the Q60 designer.

When it comes to the Aurora-type displays it is much more joy and light. Get a hold of a version of SMSQ/E for Aurora and Gold Card and you can enjoy the delights of 256 colour (also known as 8-bit colour) modes as you would on an Aurora or recent versions of QPC2, which also has an Aurora-compatible mode.

The easiest way to see this in action is to put a copy of SMSQ_GOLD for Aurora (note: it must be an Aurora version, versions for standard QL and Gold Card may not work correctly for this purpose, although they will run on QemuLator in simple mode 4 and mode 8) on a disk to transfer to QemuLator. For the purposes of this article, I used version 3.13 of SMSQ/E for Aurora+Gold Card. You can check the SMSQ/E version with the command

```
PRINT VER$(1)
```

QemuLator can only display 256 colours (mode 16) in a resolution of up to 512x480 pixels (or 512x256 if you prefer). In standard mode 4, it can manage resolutions of 512x256, or 1024x768.

Before we start QemuLator, we need to make sure it is set up for Aurora video card mode. Fire up QemuLator, and go into the "QL" menu and select the "Configuration" command and when the configuration screen appears, click on the "Graphics" tab (see figure 1). Beside the words "Additional video card" is a drop down list which offers a choice of None, Aurora, or Q60. Select the Aurora option as shown in figure 1. For the moment, it doesn't matter whether you choose the "Use accelerated QDOS graphics routines" or not.

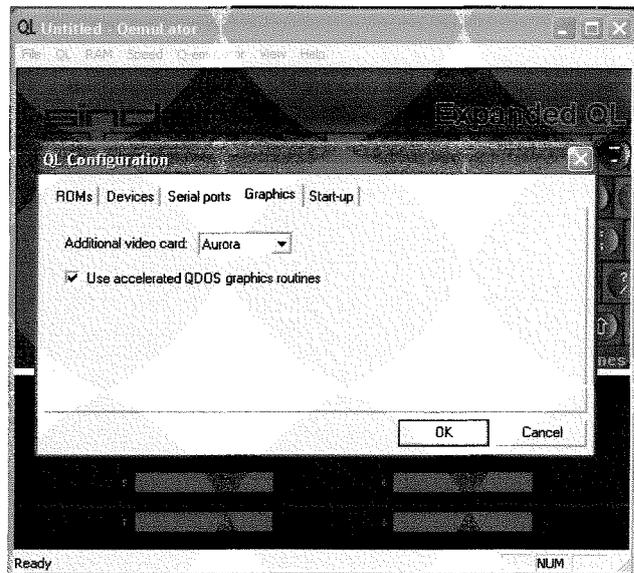


Figure 1 - Setting the video card graphics mode

As your copy of SMSQ_GOLD is on floppy disk for now, make sure that one of QemuLator's 8 drives is pointing at the floppy disk drive a: and another is pointing at your usual directory for one of your WIN drives - see figure 2. If you are unsure how to set this up, just click on one of QemuLator's 8 drive slots, which brings up the menu offering the following 4 options, from where you can choose which drives or directories or QXLWIN file is attached to the available drive slots.

- Attach directory..
- QDOS floppy disk ->
- QXL hard disk file
- Empty

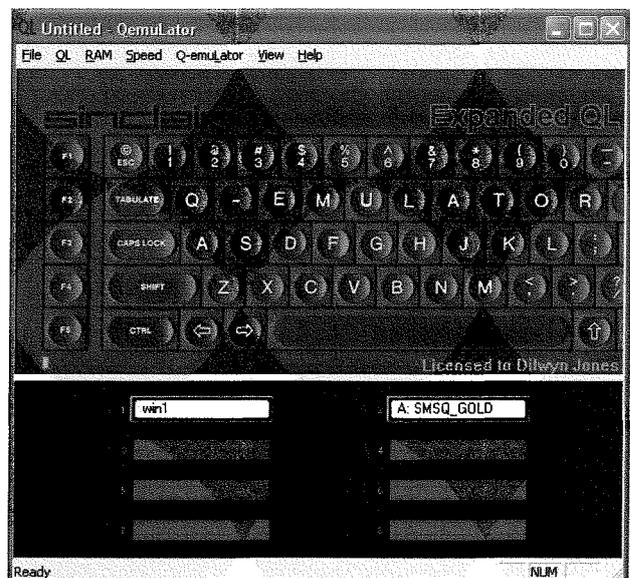


Figure 2 - drive assignment

For the purposes of this example, I'll assume WIN1_ is tied to slot 1, while the floppy disk drive is drive 2.

Either way, we need to copy files from the floppy disk to the hard disk so that SMSQ_GOLD will be available to us in future.

Click on the picture of the QL in QemuLator to start the emulation. Pressing F1 or F2 starts the QL going in the same way as it would a standard QL.

Now copy the SMSQ_GOLD file onto the hard disk for future use:

```
COPY FLP2_SMSQ_GOLD to win1_SMSQ_GOLD
```

Also, copy the MENU_REXT file to flp1_. This is needed by MenuConfig.

You may want to copy MenuConfig as well so that you can configure the SMSQ, but you can't run that yet unless you have pointer environment installed.

Now type in the command:

```
LRESPR FLP1_SMSQ_GOLD
or
EXEC FLP1_SMSQ_GOLD
```

if you don't have the Toolkit 2 command LRESPR on your system.

There will be a brief pause while QemuLator starts SMSQ/E, then the familiar red and white QL windows should appear, filling the whole PC screen. With luck, we are now in SMSQ/E and can start to set ourselves up for 256 colours.

The first thing to do is to check what colour mode SMSQ/E has set to start with. Enter the command:

```
PRINT DISP_TYPE
```

If this prints 4 or 8, you have started in plain old mode 4 or 8 (4 or 8 colour mode). If this says 16, you are in Aurora 256 colour mode. If it says anything other than 16 we need to persuade it to change to mode 16 with the DISP_COLOUR command:

```
DISP_COLOUR 2
```

Now, assuming we have the correct versions of SMSQ_GOLD, PRINT DISP_TYPE should show a value of 16 to indicate that we are in mode 16 - the 256 colour Aurora mode. We may not necessarily be in the correct colour palette mode, we may find that colours 0 to 255 are still the stippled mode 4 type colours. Here's one way of checking - enter and run this short program (warning: if you are prone to epilepsy, increase the

PAUSE value as quite a lot of screen flickering and flashing is to be expected):

```
100 FOR a = 0 TO 255
110 PAPER a
120 CLS
130 PAUSE 10
140 END FOR a
```

If you see stippled colours, you are in the QL colours palette. We need to change to the 256 colour palette mapped definition with the command COLOUR_PAL. Running the little BASIC program above once again should now show various shades of the usual 8 QL colours.

It is possible to configure SMSQ_GOLD to start in this colour depth if you wish, or you can put commands into your boot program to set the display type you require. To reconfigure SMSQ/E execute the MenuConfig program. Ensure menu_rext has been installed first:

```
LRESPR WIN1_MENU_REXT
EXEC MenuConfig
```

and follow the prompts to reconfigure SMSQ_GOLD.

It is quite possible to write a boot program which lets you choose whether to start QemuLator in traditional QDOS or in SMSQ/E. Here is an example, based on testing for VER\$ = 'HBA' which indicates SBASIC is present, in other words, SMSQ/E.

```
100 IF VER$ = 'HBA' THEN
110 REMark SMSQ/E loaded
120 PRINT 'SMSQ/E version ';VER$(1);'
    loaded...setting mode 16'
130 DISP_COLOUR 2 : REMark set mode 16
140 COLOUR_PAL : REMark 256 colour
    palette
150 ELSE
160 CLS : PRINT '1 - QDOS'
170 PRINT '2 - SMSQ/E'
180 k$ = INKEY$(-1)
190 IF k$ = '2' THEN
200 EXEC win1_smsq_gold : REMark
    start SMSQ/E
210 ELSE
220 CLS : PRINT 'QDOS'
230 END IF
240 END IF
```

Line 100 tests to see if SBASIC is present, which indicates we are running in SMSQ/E. If it is, lines 130 and 140 set mode 16 and the 256 colour palette respectively. If you wish, you can also use DISP_SIZE to set the preferred display resolution - the maximum is 512x480 in mode 16:

```
145 DISP_SIZE 512,480
```

If you wish to remain in mode 16, but wish to switch to the QL colours temporarily, you can opt to use the QL colour definitions with the command COLOUR_QL, which will use QL colours 0 to 7 and the stipples thereof.

You will find that QemuLator always goes into full screen mode when using SMSQ_GOLD. You can use ALT TAB in Windows to switch in and out of Windows to other applications as required. If you need to return to QemuLator's menus (e.g. to change one of the 8 drive assignments) you can press F12 to toggle between full screen and QemuLator menus, but it does tend to kill SMSQ/E, so you have to go into the "QL" menu, and use the "STOP" command to stop the emulation, then restart it in the usual way. I'm not sure if

this is a bug or just a limitation in how QemuLator can work.

QemuLator can be downloaded as a shareware program from Daniele Terdina's website at: www.terdina.net/ql/q-emulator.html

The free trial version limits itself to running at QL speed and with only 128K or 384K RAM. This version does not run SMSQ/E at all. Register with the author (registration costs 40 U.S. dollars) and you get a code to convert it to a much upgraded version with many extra facilities such as level 2 subdirectories, access to QXLWIN files, TCP/IP driver, faster loading of BASIC programs and much faster operation and support for much larger RAM.

Executable Things - Part 2

by George Gwilt

Previously I described the Thing THING. This is the last Thing in the Thing's linked list. Its purpose is twofold. First it allows those who are not using the operating system SMS2 (and I suspect that is most of us) to use SMS2's extra Trap #1 routines. These routines all relate to Things. I have already mentioned the one used to link a new Thing into the list.

The second purpose of THING is to enable the execution of an executable Thing. I found the need to investigate this routine for the following reason. An executable Thing will normally be started by the keyword EXEP. This is the equivalent of starting an executable program by EX. Thus I can type:

```
EX NET_PEEK
```

or

```
EXEP NET_PEEK
```

to achieve what, to all intents and purposes, is the same result.

However, EX allows a user to put information on the program's stack when it is started. So:

```
EX NET_PEEK; "Hullo NET_PEEK"
```

would set the string "Hullo NET_PEEK" onto NET_PEEK's stack.

So would:

```
EXEP NET_PEEK; "Hullo NET_PEEK"
```

So far so good.

EX goes further though. It allows channel ID's to be put on the stack too.

At this point I should describe what this looks like to a programmer.

From the earliest days of the QL every job created by MT_CJOB had its stack pointer reduced by 4 from its highest position. Provided that the dataspace requested was at least 4 this sets two zero words on the stack. It was envisaged that when each program started it would have on its stack a word giving the number of channels followed by that number of IDs. After that would be a word giving the length of a following string. The two zero words of course signify no channels and no string. When TK2 appeared it contained EX which replaced the original EXEC and allowed both channels and a parameter string to be set.

Thus:

```
EX NET_PEEK, ram1_data, #3, #4; "Hullo NET_PEEK"
```

would put on NET_PEEK's stack 3 followed by three IDs then 14 followed by the string. The first ID would be for the channel opened for ram1_data with NET_PEEK as the owner. The owner of the other channels, #3 and #4, would be the BASIC from which the EX command had been given. When I tried this with EXEP I discovered that it allowed no channels of any sort. At this stage I decided to investigate what was probably at the back of EXEP. That is the second vector routine of THING, TH_EXEC.

Here is what TH_EXEC says it expects in its registers:

- D1.L The ID of the owner
- D2.L $\text{priority} * 2^{16} + \text{timeout}$ (0 for 'EX' or -1 for 'EW')
- A0 pointer to the Thing's name
- A1 pointer to the parameter string

At first sight this is not encouraging for one hoping to set channels IDs on the Thing's stack. However, investigation showed that the "parameter string" was actually the entire contents of the stack. That is the string contained the number of channels followed by the IDs then the parameter's length followed by the parameter. So TH_EXEC can put channel IDs to an executable Thing's stack after all.

By now I had determined to produce a keyword EXEG which I would use in place of EXEP when I wanted to set channel IDs on a Thing's stack. I would type:

```
EXEG NET_PEEK, ram1_data, #3, #4; "Hullo NET_PEEK"
```

to put the IDs and string on the stack.

I would have to write code to examine the parameters of EXEG and from this make up the contents of the Thing's stack to be presented to TH_EXEC as the "parameter string" to which A1 points.

There is a snag, however. When EX produces the ID for a file, such as ram1_data in my example, it sets the owner of the ID as the program it is executing. Since the IDs have to be produced before TH_EXEC is called this would not be possible with EXEG.

There is at least one way round this difficulty. The real reason why a channel ID should have the executable Thing as owner is that the channel will automatically be closed when the job stops. It would be possible to have an intermediate job which would both be the owner of the channel and would also call TH_EXEC. If this intermediate job removed itself when the executable Thing did, then the effect would be as if the Thing were the channel's owner.

Well, yes, this works but is rather messy. In the end I decided to forget all about TH_EXEC and tackle the problem directly.

The general idea was to create the Thing's job, then examine the parameters of EXEG and load the result onto the Thing's stack. But before we create the job we need to know the number of bytes which the channel IDs and the parameter string will need, because this has to be added to the dataspace defined in the executable Thing.

There are nine steps. The code for these will be given later but it is worth a preliminary comment on each of them here.

1. Find the number of channels and length of parameter string

We have to find this information from the parameters presented to EXEG. Information about all parameters is set in 8-byte blocks. The first of these is found at (A3,A6.L) and the end of the last at (A5,A6.L). These addresses point to the last section of the Name Table in which the parameter information is stored.

The number of parameters can be calculated as

$$(A5 - A4) / 8$$

We need to know the format of the 8-byte entry for a parameter. It is:

Word	Word	Long
Type	-> Name Table	-> Value

Type

Type is a mixture of a code giving the type of parameter, a code giving the type of separator which follows the parameter and a third code indicating whether or not the parameter is preceded by hash (#).

The second of the three codes is a number from 0 to 7 held in bits 4 to 6 of the Type word. All we need to know is that number 1 is a comma (,) and number 2 is a semicolon (;).

The third code is in bit 7. Zero means no hash and 1 means there is a hash.

The remaining bits in the Type word which indicate the type of parameter include:

\$0002 for undefined floating point and

\$0201 for string variable

-> Name Table

The second word is a number giving the entry in the Name Table of the parameter's name or -1 if there isn't an entry. The nth entry will be indicated by the number n-1.

It is important to realise that this applies only to parameters. A normal Name Table entry also has a pointer in this position, but its value is the offset to the name in the Name List.

Jan Jones says in QL SuperBASIC - The Definitive Handbook,

"... an entry is made at the top of the nametable for all the actual parameters. Such entries are not permanent, they only exist for the duration of the procedure. If the parameter is a simple variable, the new entry is a copy of the entry for that name with the pointer to the namelist replaced by a pointer to the original nametable entry."

-> Value

If there is a value for the parameter it can be found from the long word pointer which is the offset from the start of the variables area.

The parameters presented to EXEG consist of the Thing name, which must be present, followed by the channels and then the parameter string for the Thing itself. If t is the number of parameters following the Thing name, then the number of channels is t if there is no parameter string and t - 1 otherwise. We can tell if there is a parameter string by examining the second last separator, if t > 0. If t = 0, then there are no channels and no parameter string. If the second last separator is a comma there is no parameter string. If the separator is a semicolon there is a parameter string.

If there is a parameter string, its length is found by examining its value in the variables area.

2. Find the Thing's Name

The first parameter to EXEG is the Thing's name. This might be presented with or without quotes. If it is without quotes it will be taken by the system as an undetermined floating point variable. In the first case its type code will be \$0201 and in the second \$0002.

In the first case the name can be put onto the maths stack by the vector CA_GTSTR. In the second case the name has to be extracted from the Name List and put on the maths stack. This is done by the subroutine n_to_stack. This finds the position in the Name Table for the name we want. In this case the second word in the 8-byte entry gives the offset in the Name List of the name we want.

After a check that there is enough room on the maths stack, by using BV_CHRIX, the routine copies the name to the stack from the Name List.

3. Find the Thing's Linkage Block

The Thing's linkage block is found by scanning the chain of linkage blocks until one is found with the Thing's name. There is a special subroutine, cp, which compares the Thing's name, which is on the maths stack, with the name in the linkage block. We could have used the vector UT_CSTR but this would have entailed subtracting A6 from A2 since both A1 and A2 have to be relative to A6 and only A1 is. This is unsafe since A6 can change at any time. So I wrote the subroutine cp instead.

4. Find the Thing

The address of the Thing is at long word \$10 from the start of the linkage block.

5. Create the Thing's Job

We need to set:

D1.L = the owner job : We set that to 0 for "independent"
D2.L = length of code : From the Thing at \$C(A0)
D3.L = dataspace : From the Thing at \$10(A0) plus space for the channel
; IDs and parameter string
A1.L = start address : From the Thing at \$14(A0)

6. Adjust the Thing's stack and add the code

When the job is successfully created we first adjust the job's stack to make room for the IDs and string. The address in A0 set by MT_CJOB is the address immediately following the \$68 byte header set up for the job. The position for A7 in this header is thus at -12(A0)

We then fill in the code from the address given in the Thing to the address in A0 set by MT_CJOB.

7. Put the channel IDs on the stack

The number of channel IDs to be put on the stack is in the top word of D5 (nc). This is set on the stack. We then examine nc parameters and set the appropriate IDs on the stack. For each parameter there are two possibilities. Either the channel is explicit, heralded by hash (#) or it is the name of a file which must be opened. We determine whether it is hash or not by testing bit 7 of the second byte of the parameter block. If this bit is set we read in the integer parameter and use it to access the BASIC channel block which contains the ID we want.

Each BASIC channel block is \$28 bytes long and the first one is at the address BV_CHBAS(A6) relative to A6. The ID is the first long word in the block.

If we find that there is no hash we examine the type of parameter to see whether it is a proper string by testing the first byte of the parameter information.

This will be non zero for a string. Actually the type word should be \$0201 for a string and \$0002 otherwise. If the type is in fact something else we rely on an error being signalled when we attempt to put the string on the maths stack.

When the string has successfully been read in we try and open the file with the name given. If this has not worked, we try again having added DATAD\$ to the start of the string. Note that we set D3 = 0 for the OPEN. This is the "old exclusive" form of OPEN which is how EX opens files. The owner of the channel is the job we have just created. Again this is what EX does.

DATAD\$ is added by the subroutine ad_dat. This checks that there is space on the maths stack for the addition of DATAD\$. It then subtracts the rounded up length of DATAD\$ from the maths stack and places DATAD\$ there. If DATAD\$ is of even length the new filename is set. Otherwise we must move the old filename back by one byte so that it is properly joined up with DATAD\$.

8. Put the Parameter list on the Stack

The rounded up length of parameter list to be set on the stack is now in the top word of D5. If this length is zero there is no parameter.

If there is a parameter we ignore its type code and attempt to set it on the maths stack by CA_GTSTRG which will signal an error if it is not a string.

9. Activate the job

The job is activated by a call to MT_ACTIV which requires the job ID to be set in D1, the priority in D2 and timeout in D3.

We set priority to 16 and timeout to 0, which corresponds to EX rather than EW.

The Code

Here is the code which will set EXEG as a keyword which is capable of executing executable Things with a set of channel IDs and a parameter string on its stack.

The syntax is

```
EXEG tname(, filename or #channel)[;parameter string]
```

tname is the Thing name (with or without quotes or apostrophes)

() means optional repeated

[] means optional once only

The Listing

```
; exeg4_asm
    IN          WIN1_LIB_FNPROC_ASM macro for defining procs and funcs
    IN          WIN1_LIB_HED1      macro for setting strings

sys_thgl equ    $b8      address of start of Thing linkage blocks
bv_ntbas equ    $18      base of Name Table
bv_ntp  equ    $1c      current top position in Name Table
bv_nlbas equ    $20      base of Name list
bv_vvbas equ    $28      base of variables area
bv_chbas equ    $30      base of basic channels
bv_chp  equ    $34      current top of channels
bv_rip  equ    $58      maths stack

*****
*   The following code links in EXEG as a keyword using LRESPR   *
*****

START    LEA      DEFINE,A1
         MOVEA.W  BP_INIT,A2
         JMP      (A2)

define
proc_start
pf_name  exg,EXEG  The code for EXEG is at exg
proc_end
fn_start
fn_end

*****
* 1. We first find the number of channels (nc) and the length of *
*      the parameter string rounded (lc).                        *
*****

exg      moveq    #1,d7          lc = 0 (+ 1 for rounding)
         moveq    #-8,d5
         add.l    a5,d5
         sub.l    a3,d5
         asr.l    #3,d5          t (number of pars - 1)
         bmi     bad_parm  ----> must be > 0 pars
         beq     exg_2       no parameters for the Thing
         bfxetu  -15(a5,a6.1){1:3},d0 second last separator
         subq.w  #1,d0        comma? . .
         beq     exg_2       . . yes (no parameter string)
exg_1    subq.w  #1,d0        semicolon? . .
         bne     bad_parm  ----> . . no
         subq.w  #1,d5        nc = t - 1
         move.l  -4(a5,a6.1),d0 offset to string
         add.l   bv_vvbas(a6),d0
         add.w   (a6,d0.1),d7   lc = par length (+ 1 for rounding)

; D5 = nc: D7 = lc + 1
*****
* 2.          Now get the Thing name                            *
*****
```

Q U A N T A



Independent QL Users Group

World-wide Membership is by subscription only,
offering the following benefits:

Bimonthly Newsletter – up to 40 pages

Massive Software Library – All Free!

Free Helpline and Workshops

Regional Sub-Groups. One near you?

Advice on Software and Hardware problems

Subscription just £14 for UK members

Overseas subscription £17

Barclaycard: Visa: Access: MasterCard: Accepted

Now in our Twenty Fourth Year

Further details from the Membership Secretary

**John Gilpin, 181, Urmston Lane
Stretford, Manchester, M32 9EH (UK).**

Tel. +44 (0) 161 865 2872

Or

Visit the New Quanta Web Site

<http://www.quanta.org.uk>

E-mail: membership@quanta.org.uk

```

exg_2    bclr      #0,d7          lc rounded up to even
         move.w    (a3,a6.l),d0   1st word of par info
         andi.w    #$ff0f,d0     strip separator
         cmpi.w    #$201,d0      string variable? . .
         beq      exg_3          . . yes
         subq.w    #2,d0         must be this? . .
         bne      bad_parm ----- . . it isn't!!
         bsr      n_to_stack     get name from name list to maths stack
         bne      bad_parm -----
         bra      exg_4
exg_3    bsr      gstring        get ordinary string
         bne      bad_parm -----

```

; The Thing's name is on the maths stack

```

*****
* 3.      We must find the Thing's linkage block *
*****

```

```

exg_4
         moveq     #mt_inf,d0      System information
         trap      #1             AO -> system variables
         move.l    sys_thgl(a0),d0 Thing list
         bra      thok
thok1   move.l    (d0.l),d0        next thing linkage block
thok    beq      bad_parm ----- either no list or not found
         lea      $2a(d0.l),a2     -> name
         bsr      cp              compare names
         bne      thok1          not this one

```

```

*****
* 4.      We must locate the Thing *
*****

```

```

         movea.l   $10(d0.l),a0    -> thing
         move.l    4(a0),d0        type
         subq.l    #1,d0           is it executable? . .
         bne      bad_parm ----- . . no

```

; AO -> Thing

```

*****
* 5.      We must create this as a Job *
*****

```

```

         movea.l   8(a0),a2
         adda.l    a0,a2           -> code
         move.l    $C(a0),d2      code length
         move.l    $10(a0),d3     dataspace . .

```

; We need to increase the dataspace by $nc*4 + lc + 4$

```

         move.l    d5,d0          nc
         asl.l     #2,d0          nc*4
         swap     d5
         move.w    d7,d5          D5 = nc | lc
         add.l     d0,d7          nc*4 + lc
         add.l     d7,d3          adjusted dataspace
         addq.l    #4,d3          4 bytes for the counts
         move.l    $14(a0),d0     Program start
         beq      exg_5          A1 -> 0

```

```

add.l    a0,d0          A1 not zero
exg_5    movea.l  d0,a1

; We now create the Job

moveq    #0,d1          independent job
moveq    #mt_cjob,d0
trap     #1
tst.l    d0
bne      bad_parm  ---->  can't start a job!!
move.l   d1,d6          keep ID in D6.L

```

```

*****
* 6.      Adjust the stack and fill in the code          *
*****

```

```

movea.l  -12(a0),a4      stack to A4
suba.l   d7,a4          adjusted stack for chans and par
move.l   a4,-12(a0)     replace stack pointer
lsr.l    #1,d2          code length in words
bra      exg_6

```

```

; There may be more than 2^32 bytes of program so we count the
; two halves of D2 separately

```

```

exg_7    swap    d2
exg_8    move.w   (a2)+,(a0)+    copy code a word at a time
exg_6    dbf     d2,exg_8
         swap    d2
         dbf     d2,exg_7

```

```

; Now A4 -> prog's stack for filling
; D5 = nc (number of channels) | lc (length of par string)
; D6 = Job ID for Thing
; A3 -> parameters after the Thing's name

```

```

*****
* 7.      Find the channel IDs and put them on the stack  *
*****

```

```

swap     d5              D5.W = nc
move.w   d5,(a4)+        count of channels
bra      ta4

ta10     btst    #7,1(a3,a6.l)  is it #? . .
         beq     ta5              . . no
         bsr     gtin            get channel number . .
         bne     nochan  ---->
         move.w  (a1,a6.l),d1     . . to D1
ta9      move.l  bv_chbas(a6),a0
         mulu.w  #$28,d1
         add.l   d1,a0           -> channel
         cmpa.l  bv_chp(a6),a0
         bge     nochan  ---->  over the top!
         move.l  (a0,a6.l),d1     ID
         bmi     nochan  ---->  gone!!
         movea.l d1,a0           ID to A0
         bra     ta6

ta5      tst.b   (a3,a6.l)      Not a string type
         beq     ta7

```

```

        bsr      gstring      String to maths stack
        bne      bad_parm3 ---->
        bra      ta8
ta7     bsr      n_to_stack   Name to maths stack
        bne      bad_parm3 ---->
ta8     bsr      ope          try and open the file
        beq      ta6          OK
        bsr      ad_dat       add DATAD$ . .
        bne      nochan ---->
        bsr      ope          . . and try again
        bne      nochan ---->   didn't work

ta6     move.l   a0,(a4)+     set ID on stack

ta4     dbf      d5,ta10     count channels

```

```

*****
* 8.          Put the parameter string on the stack          *
*****

```

```

        swap     d5          D5.W = 1c (rounded)
        tst.w    d5          par string? . .
        beq      ta11       . . no
        bsr      gstring
        bne      bad_parm3 ---->
        move.w   (a1,a6.1),d0 length of par string . .
        move.w   d0,(a4)+   . . set on stack
        bra      ta12
ta13    move.b   2(a1,a6.1),(a4)+ copy the string to the stack
        addq.l   #1,a1
ta12    dbf      d0,ta13

```

```

*****
* 9.          Activate the job                               *
*****

```

```

ta11    move.l   d6,d1       ID of Thing job
        moveq    #$10,d2     priority 16
        moveq    #0,d3       EX (zero timeout)
ta14    moveq    #mt_activ,d0
        trap     #1
        rts              return to BASIC

```

; Sub Routines

; n_to_stack sets the name list entry for (a3,a6) to the maths stack
; uses no reg except that A3 is updated to point to the next parameter

```

ns_reg  reg      d0-3/a0/a2
n_to_stack
        movem.l  ns_reg,-(sp)
        movea.l  bv_ntbas(a6),a0
        move.w   2(a3,a6.1),d0   index to name
        lea      (a0,d0.w*8),a0  -> name table entry
        cmpa.l   bv_ntp(a6),a0
        bcc      nts1 ---->
        move.w   2(a0,a6.1),d1   offset to name list
        bmi      nts1 ---->
        movea.l  bv_nlbases(a6),a0
        add.w    d1,a0           -> name in name list
        moveq    #3,d1

```

```

        add.b      (a0,a6.1),d1      name length + 2 . .
        bclr      #0,d1              . . rounded up
        move.w    d1,-(a7)           length needed on maths stack . .
        movea.w   bv_chrix,a2
        jsr       (a2)               . . ensure it is there
        movea.l   bv_rip(a6),a1      maths pointer
        suba.w    (a7)+,a1           point to start of string . .
        move.l    a1,bv_rip(a6)     . . and update stack
        moveq     #0,d1
        move.b    (a0,a6.1),d1      name length to D1.W
        move.w    d1,(a1,a6.1)
        bra       nts3
nts4    addq.l    #1,a0               advance A0
        move.b    (a0,a6.1),2(a1,a6.1) -> stack
        addq.l    #1,a1               advance A1
nts3    dbf      d1,nts4
        movea.l   bv_rip(a6),a1      reset A1 to start
        addq.l    #8,a3              -> next parameter
        moveq     #0,d0
nts2    movem.l  (sp)+,ns_reg
        rts
nts1    moveq     #-1,d0
        bra       nts2

```

```

; gstring and gtin get one string and one integer parameter.
; If an error occurs condition code is NE.
; A3 is updated to the next parameter.
; No registers are used except A2

```

```

gs_reg   reg      d0-4/d6/a0/a4
gtin     movea.w   ca_gtint,a2
        bra       gs1
gstring  movea.w   ca_gtstr,a2
gs1      movem.l  gs_reg,-(sp)
        movea.l   a5,a4              keep A5
        lea      8(a3),a5           set for 1 parameter . .
        jsr      (a2)               . . and get it
        movea.l   a5,a3              update A3
        movea.l   a4,a5              replace A5
        movem.l  (sp)+,gs_reg
        rts

```

```

bad_parm3 bsr      remv
bad_parm  moveq    #-15,d0           bad parameter
        rts

```

```

nochan   bsr      remv
        moveq    #-6,d0           invalid channel
        rts

```

```

remv     move.l    d6,d1             ID of Thing job
        moveq    #0,d3             No error
        moveq    #mt_frjob,d0      force remove job
        trap     #1
        rts

```

```

; cp compares a string at A2 with a string at (A1,A6.L) regardless of case.
; Returns cc EQ if found else NE. No regs used.
; NOTE. The string length must not be zero.

```

```

cp_reg    reg        d0-2/d3/a1-2
cp        movem.l    cp_reg,-(sp)
          move.w     #$DF,d3           prepare for ANDing
          move.w     (a2)+,d0         length
          cmp.w      (a1,a6.l),d0     same? . .
          bne       cp_end           . . no
          subq.w     #1,d0
cp1       move.b     (a2)+,d1
          move.b     2(a1,a6.l),d2
          eor.b      d2,d1           set bits which differ
          and.w      d3,d1           ignore case
          addq.l     #1,a1           cond codes unaltered
          dbne      d0,cp1          carry on till unequal
cp_end    movem.l    (sp)+,cp_reg
          rts

```

```

; ad_dat adds DATAD$ to the start of the name at (A1,A6.L)
; A1 is updated and set to BV_RIP(A6)
; On error D0 is set to "not found". CC are set
; no other registers are used

```

```

reg_dat   reg        d1-4/a0/a2
ad_dat    movem.l    reg_dat,-(a7)
          moveq      #mt_inf,d0       System information
          trap       #1              A0 -> system variables
          move.l     sv_data(a0),d0   -> DATAD$
          beq       not_found ----->
          movea.l    d0,a0           Pointer in A0
          moveq      #0,d1
          move.w     (a0)+,d1        length to D1.L for BV_CHRIX
          move.w     d1,d4           keep in D4.W
          move.l     a1,bv_rip(a6)   store current value of stack
          movea.w    bv_chrix,a2     check there's room
          jsr       (a2)
          movea.l    bv_rip(a6),a1   new position (possibly)
          move.w     (a1,a6.l),d3    original length of filename
          movea.l    a1,a2           keep position in A2
          suba.w     d4,a1           new value of A1 . .
          btst      #0,d4           odd? . .
          beq       ad_dat4         . . no
          subq.l     #1,a1          set back to even address
ad_dat4   move.l     a1,bv_rip(a6)   store new position
          move.w     d3,(a1,a6.l)    set new . .
          add.w      d4,(a1,a6.l)    . . length
          addq.l     #2,a1           skip length
          bra       ad_dat5
ad_dat6   move.b     (a0)+,(a1,a6.l) insert DATAD$
          addq.l     #1,a1
ad_dat5   dbf       d4,ad_dat6
          cmpa.l     a1,a2           are we at the same place? . .
          beq       ad_dat7         . . yes
          bra       ad_dat1
ad_dat2   move.b     1(a1,a6.l),(a1,a6.l) move old name . .
          addq.l     #1,a1           . . back . .
ad_dat1   dbf       d3,ad_dat2     . . one byte
ad_dat7   movea.l    bv_rip(a6),a1   reset A1
          moveq      #0,d0
ad_dat3   movem.l    (sp)+,reg_dat
          rts

```

```

not_found moveq    #-7,d0
          bra      ad_dat3

ope       moveq    #0,d3          OPEN (old exclusive)
          move.l   d6,d1          ID
          trap     #4            A1 relative to A6
          movea.l  a1,a0          -> name (rel to A6)
          moveq    #io_open,d0
          trap     #2
          tst.l    d0
          rts

```

Unusual Instructions

The above code uses three types of instruction which requires a 68020+ for its execution.

a. While searching through the linkage blocks for the Thing's name the instructions

```

thok1    move.l   (d0.l),d0      next thing linkage block and
          lea     $2a(d0.l),a2   -> name

```

appear.

The effective addresses (d0.l) and \$2a(d0.l) have no base register, thus allowing a data register to be used as if it were an address register.

b. A bit field instruction is used to extract a separator towards the start of EXEG. Such instructions allow a contiguous set of bits from 1 to 32 bits to be manipulated. The start of the set of bits is defined by an offset from an effective address. The size of the bit field can be set to any number from 1 to 32. If a data register is used to define the offset values from -2^{31} to $2^{31}-1$ can be set. Otherwise the offset is restricted to a number between 0 and 31.

The instruction used here is:

```

bfxetu  -15(a5,a6.l){1:3},d0 second last separator

```

The offset is 1 and the size 3 as you can see from the numbers inside the curly brackets. bfxetu stands for Bit Field EXtract Unsigned. The three bits extracted are placed in the bottom three bits of D0. The remaining bits of D0 are set to zero.

c. In the subroutine n_to_stack we need to locate the n - 1th entry in the Name Table. Given the number n - 1 in D0 we require eight times D0 to be added to the base address of the table. The instruction used is:

```

lea    (a0,d0.w*8),a0 -> name table entry

```

You can see that the index register D0 has been multiplied by 8. Values of 2, 4 and 8 are allowed.

Negotiation, Negotiation, Negotiation

by Geoff Wicks

I had a moment of absolute horror when I saw the hotel room for the first time. It was just 8.5 metres by 4.5 metres. Even the hotel said the maximum capacity was 30 people, and that was with all of them seated theatre style. There was no way we could hold a show in that room. All I

could do was to wait for the Quanta Committee to arrive and take it from there.

I went for breakfast and was allocated a table next to a familiar face. John Mason was unconcerned about the problem, but then he had done

his homework the night before. He was confident the show would go on.

We had booked a room much larger, but the hotel had had to change the arrangements because another, later, booking required a room with a bar. As it happened this turned out to be to our advantage, but only after numerous negotiations between myself, traders, speakers, Quanta and the hotel. Goodwill ruled the day and we ended up with a mini conference centre all to ourselves.

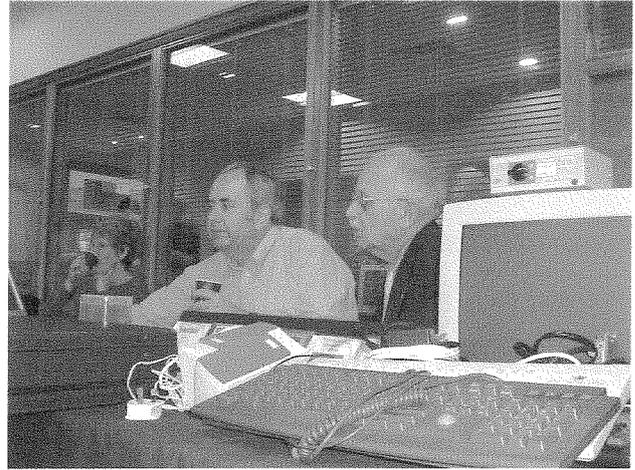


John Mason and I had a leisurely breakfast and, by the time we had finished, the traders had already claimed their territory. Quanta was banished to the central hallway of the conference centre where we had discovered more electric sockets and a supply of tables. It turned out to be a shrewd move because that was where the free coffee and tea machines were and it soon became the natural centre of gravity of the show.



We had scarcely finished setting up before the first punters arrived, and with them the first signs that this show was going to be different. It is over 10 years since the last Midlands workshop and there were numerous faces we did not recognise, plus some more familiar ones from the past. Making a welcome comeback to a QL show were two traders, Rich Mellor and Derek

Stewart. In addition Roy Wood and Tony Firshman were present together with semi-traders Dilwyn Jones and Geoff Wicks.



Among the unfamiliar faces were a couple of people who had left the QL scene, but had come along to see what was currently happening in the QL community. Quanta gained two new members and old acquaintanceships were renewed with QL-ers from the Midlands. The official Quanta statistic, systematically researched, was that 32 people attended the workshop.

Crisis came just after eleven o'clock when our main presenter, Simon Goodwin, arrived. He took one look at the available space and announced he could not give his presentation. More hurried negotiations took place as a result of which the hotel agreed to give us another room for a peppercorn fee. Simon and entourage took over this room and posted a guard on the door so that he could set up without interruptions. This was necessary as three weeks before the show Simon had enquired about the possibility of an internet link and other technical matters, but Quanta had failed to reply.



As Simon was using non-standard equipment we knew there could be a problem with the computer projector. John Mason had brought a pro-

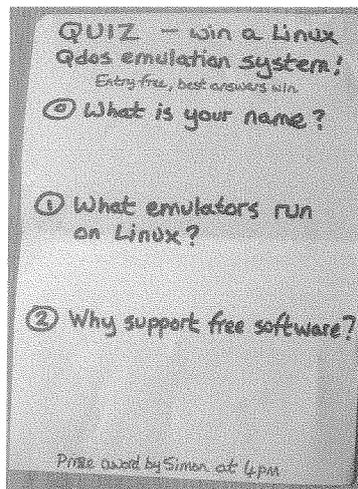
jector with him, but as the conference rooms had plasma screens that also served as monitors, he thought these would be better. Simon and Tony Firshman went to work on the technical side and soon discovered there were problems with un-screened cables. Another round of negotiations followed to secure the hotel's agreement to allow us to connect a monitor cable directly to the screen. Yet more negotiations were necessary to get Quanta's agreement to an internet link, although this had to be abandoned because Tony and Simon had insufficient time to solve all the technical issues.



Inevitably we started the presentations late. Simon had brought with him a Linux system computer that he had donated as a prize. He had set three competition questions numbered 0, 1 and 2 starting with the difficult one, "What is your name?" Then came "What emulators run on Linux" followed by "Why support free software?" The purpose of his presentation was to show how it is possible to emulate a QL on a non-Microsoft computer system using free software and cheap components.

Simon demonstrated all the possibilities of the computer including downloading images from a digital camera and displaying these via photon. He then did a MIDI demonstration using a Casio keyboard. He went on to describe and demonstrate the emulators that can run on Linux systems.

After an hour Simon was still in full flow and continued to hold the attention of



his audience for a further 30 minutes. After this epic performance I announced a break for a late lunch before continuing the presentation programme. Fortunately the hotel provided good quality bar meals at reasonable prices.



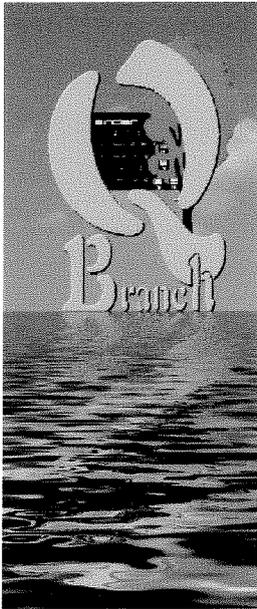
It was back to emulators after lunch when Dilwyn Jones demonstrated his "QL on a Stick", although only after another delay to solve a technical problem. Dilwyn was using my laptop, which refused to recognise his USB memory stick. Fortunately he had a spare with him.

The idea of a QL on a Stick was suggested about three years ago as a form of portable QL that could be run on almost any PC. There was a good deal of enthusiasm for the idea. Given the then high price of USB memory the suggestion was made that this would be a suitable project for Quanta to finance and develop. At the time I had suggested holding an informal meeting of interested people at the 2004 Quanta Workshop and AGM to discuss the technical and software problems, but no one volunteered. Thus Quanta could not proceed and interest in the project faded away.

When the idea of a QL on a Stick was revived earlier this year, Dilwyn decided to develop it himself. The big snag with a USB memory stick is that each PC into which it is plugged will assign it a different drive letter. Dilwyn had to ensure the software would run whatever the drive letter. The QL on a Stick is available on a CD for transfer to your own USB memory.

Dilwyn's presentation created a great deal of interest. There is a full report of the "QL on a Stick" in our news section. We also have a review in this issue together with articles on two of the emulators Dilwyn uses.

At this stage I must pause for a statutory Health and Safety warning. I do not know what primeval forces operate in the far reaches of the principal-



Feeling out on a Limb?
Reach out for QBranch
Suppliers of Quality QDOS/SMSQ products
Hardware and Software

20 Locks Hill
 Portslade
 Sussex
 BN41 2LB
 UK

Tel: +44(0)1273 386030
 Fax: +44(0)1273 430501
 Mobile: +44(0)7836 745501

email : sales@qbranch.demon.co.uk
www.qbranch.demon.co.uk

We can accept payment by
 Visa, Mastercard, Maestro and
 Switch. We also accept sterling
 cheques drawn on a UK bank,
 or payment direct to our bank:

Natwest
 Sort code 60-17-01
 Acc. 84534826
 BIC : NWBKGB 2L
 IBAN :
 GB34 NWBK 6017 0184 5348
 26



QDT
 The Ultimate QL Desktop
 v1.02
£41.50 + p&p
 £2.00 UK / 2.50 EU / £ 3.00 RoW
 Needs SMSQ/E & extended
 colours to run

EasyPTR v4
 Pts 1&2
 Major Rewrite!
£42.00
 Produce High Colour
 pointer-driven menus from
 SuperBASIC

Also on
 CD Rom

Text 87 £ 79.00
 High Colour Patch (all systems) £9.00
 ESC/P2 drivers £ 26.00 / Fountext £39.00
 Typset94 / 2488 drivers £29.00 ea.

SMSQ/E and QPC 2
 SMSQ/E for Gold Card / Atari / QXL / Q40
 Aurora High Colour Driver included with
 Gold Card Version
 Now Only **£15.00** + postage

QPC 2 v 3.33
 Now only **£ 42.00**

Upgrade Prices:
 from v 3.xx Free
 from v 2.xx £13.90
 from v 1.xx £34.00

Delivered on CD ROM
 with Manual as Adobe
 Acrobat File

QPCPrint
 Print to any Windows Installed Printer from
 Your QL Programs !
£27.00

Programs upgraded to high colour version
 QMake / WinEd / QPAC 1 / QPAC 2 / QSUP / SuQcess / Disk Mate 5
 Send old master disk and £1.60 per disk to cover costs

Free!

Paid Upgrades

QD 98 to QD 2003	£10.50
QSpread 2001 to QSpread 2003	£10.50

**Return Master Disk
 with order**

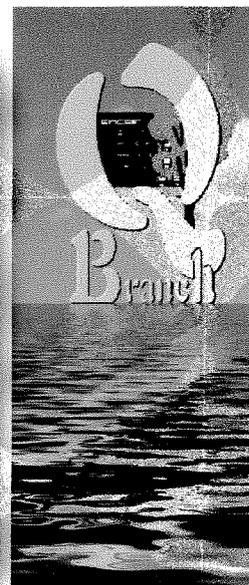
**Epson EPL-6200 Printer. QL compat-
ible Paralell Port) with 1 new toner.**

£75.00 + Carriage

Samsung 19" CRT Monitor

Sony Triniton Flat Screen

£50.00 Collection advised.



Utilities		Programs		Programming	
Fifi2	File Finder	£21.00	QD 2003	Text Editor & More	£ 49.00
QSup	Utillies	£30.00	QBASIC	QLiberater to QD Link	£ 15.00
QSpread	Spreadsheet	£51.00	QLiberator	Basic Compiler	£ 50.00
Cueshell 2	File Manager	£15.00	QD + QBASIC		£ 63.00
QPAC 2	File Manager & Utilities Package	£42.00	QD + QBASIC + QLiberator		£104.00
QPAC 1	Calendar, Clock, Calculator, Sysmon	£22.00	QPTR	Pointer Toolkit	£ 32.00
QLoad/Qref	Fast load for QDOS	£15.00	MasterSpy	Fast Text Editor	£ 30.00
QTYP2	Spell Checker	£31.00	QMake	Assembler Tools	£ 18.00
QLQ	Printer Utility	£30.00	QMon/Jmon	Monitor - Upgrade only	£ 22.00
SuQcess	Database	£28.00	BASIC Linker	Basic Library Linker	£ 22.00
Q-Route	Route Finder	£25.00	Disa 3	Dissassembler	£ 34.00
Knight Safe3	Backup Program	£35.00	QMenu	Menu Extensions & tutorial	£ 16.00
Fractal Collection	Fractals	£35.00	Easyptr v4	Toolkits & Programming Extns	£ 41.50
QCount	Accounting	£25.00	Easyptr v4	Part 3 C extensions	£ 14.00

HARDWARE

We have a rotating stock of both new and second user hardware. It is best to call or email us for details of what is available.

Recycled Items (when available)

Super Gold Card	£110.00
Gold Card	£ 45.00
Aurora	£ 65.00
QXL	£ 35.00
superHermes	£ 65.00
DiRen Keyboard	
Interface	£ 15.00
Qplane	£ 5.00

New Items

Aurora	£70.00
Aurora w/8301 & Minerva	£80.00
8mb Rom Disq	£98.00
4mb Rom Disq	£65.00
2mb RomDisq	£39.00
mPlane	£34.00
MCplate	£ 6.50
Various braQuets	£ 8.00
Gold /Super Gold Card Batteries	£10,00

We also have a collection of standard QLs, QL Power supplies and some QL books.

Cables for the Aurora, Qubide and Super Gold Card ROMs and other QL accessories are also available from us.

Call for details

ty, but they are obviously powerful. I can now confirm that all the rumours you have heard about the effect a certain Welshman has on a PC are no myths. When I arrived home and switched my laptop on for the first time after the aforesaid Welshman had touched it, it went into a continuous loop saying my system disk was invalid and that I should remove it. This is a little difficult when it is the hard disk. It is even more difficult to consult the manual when that is also on the said hard disk.

By the time Dilwyn had finished we were touching closing time and there were just 4 people at my presentation on mapping. The content will be appearing as an article in QL Today.



The final act of the show was to announce the winner of Simon Goodwin's competition. Most of the punters had gone home, but QL Today and the Quanta Magazine conspired together to set up a presentation just for the press. Maybe we were misleading our readers, but then it is not often we get the chance to print a picture of a young lady in our pages.

Quanta knew a workshop in Birmingham was going to be a high risk event with a journey into the unknown. John Mason privately expressed his surprise that no Midlands show had been held for 10 years given that his predecessor as chairman lived in the Midlands.

When we embarked on this adventure, we had little idea whether people would attend and to compound the uncertainty it had not been an easy show to organise. At the end of the day, however, we had no doubt our gamble had paid off. We had a good trader attendance, had seen several new faces and gained new Quanta members. There had been a lively "buzz" throughout the day and the presentations had attracted interest. The hotel staff had gone out of their way to accommodate our needs. There were even

vague whispers about the possible shape of a future Quanta committee. John Mason expressed his pleasure about the show and sensed a possible renewed interest in Quanta.



Perhaps, however, there are some lessons Quanta could learn. Traditionally Quanta has always left the detailed organisation of workshops to the local subgroup, and over the years this practice has worked well. Running a show is far more than just a hall with tea and coffee. The hall has to be the right size at a price Quanta can afford and have suitable electrical and catering facilities. There have to be ample tables and seating. There has to be ease of access for both car and public transport users and readily available parking so that loading and unloading is not a great problem.

The Birmingham subgroup suggested they would have found it easier if Quanta had been able to provide a document setting out guidelines for running workshops. In conversations I had with individual members of the Quanta committee they did not warm to this proposal, clearly feeling it would compromise the autonomy of the local subgroup.



Quanta is often criticised for the location of its workshops. Both Manchester and Hove are at extreme ends of the country. Would not a central inner city location be much better? Whenever

these accusations are made Quanta remains silent although it would be easy to reply to them.

The 2004 survey I did of Quanta membership indicated that over a two year period more Quanta members could attend the AGM at the present locations than if it were held each year in a central location. It should also be said that people have tried to find suitable inner city halls at a suitable price and have been unsuccessful.

Perhaps the Quanta committee need to take a greater interest in the problems of running shows so that they can defend themselves better from well meaning but ultimately unjust criticism.

But back to a central location. Quanta are planning a major show to celebrate the quarter centenary of the QL in 2009. What's the betting that the Holiday Inn in Solihull will be high on the shortlist as a possible location for this event?

Off Topic – Version Control

by Norman Dunbar

I previously did an article about using DocBook XML (ok, on a PC) to produce documentation etc in many output formats. In this article I'm going to talk about version control.

While the subject of the article is probably off topic for a QDOS/MSQ magazine, you may be pleasantly surprised to find out that it is indeed relevant – provided you use QPC or similar on a PC.

Background

In the past I have worked from home and in my lunch hour at work – indeed, that's where I am now typing this article – and I have had a separate folder on my work PC for my Assembly articles. I have another one at home for the copies and stuff I do there.

Now, in the past I have managed to trash a pile of work when I managed to copy the wrong folder to the other wrong folder, or, when I had updates in both folders and did something stupid like copying everything from one to the other and over wrote some files. Oops!

Version control, in my case SubVersion, freely

available from <http://subversion.tigris.org> – helps me to avoid the stupidity of me copying (or rsync-ing) files in the wrong direction and losing updates. There's a free PDF book available from a link on the above page – highly recommended.

My working setup

I have a work PC running Windows, a home PC running Windows and Linux and a USB drive – which is a proper hard drive in my case, but this works on USB 'stick drives' as well, if you have enough room on it. Both PCs have Subversion installed and on the path.

I have a folder (c:\source\qdos) on both my work PC and also on my home PC (under Windows) and my various files live under those locations – either in separate folders or just under the main folder itself.

On the USB drive, which is disc F: at home and J: at work – this is not a problem – I have a folder named <drive>\SVN\qdos which is my repository and holds the history etc of the changes made to the various source files.

Basic version control

The first step is to have some files that you want to put into version control. In my case, I have my previous articles and code on my USB stick. I copied the entire folder to a temporary folder on my home PC's C: drive – c:\stuff.

Next, I plugged in my USB hard drive and as ever it turned up as my F: drive. I have this configured under "Start → Settings → Control Panel → Administrative Tools → Computer Management → Storage → Disk Management" so that it always is set to the F: drive. At work, it is always the J: drive. This is about the most useful part of Microsoft Windows that I have found.

In a DOS window, I ran the following commands to create a folder on the USB drive:

```
C:\> mkdir f:\SVN
C:\> mkdir f:\SVN\qdos
```

So far so good, I then created a Subversion repository in that folder as follows:

```
C:\> svnadmin create f:\SVN\qdos
```

When I got my prompt back, a quick dir of that folder shows a lot of files have been created:

```
C:\> dir /B f:\SVN\qdos
```

```
conf
dav
db
format
hooks
locks
README.txt
```

So, I now have an empty repository ready to load my life's work into. A quick change back to the c:\stuff folder, where my files are waiting patiently, and I enter the following command:

```
C:\> cd stuff
C:\stuff> svn import . file:///f:/SVN/qdos -m "Initial load"
Adding      part_001.txt
Adding      part_002.txt
Adding      part_003.txt
...
Committed revision 1.
```

The above simply says, run the 'svn' program to 'import' the current directory and any sub directories (the dot) into the Subversion repository found locally ('file://') at location 'f:/SVN/qdos' – note the leading slash – I also supply a commit message to show subsequent users of the repository why I created revision 1. The repository need not be local – at work I have a server that I use for many Subversion repositories, the URL in that case is SVN:// or HTTP:// – Subversion can use many different protocols to access a repository.

So, after a small delay, all my files and folders underneath c:\stuff were imported into my repository on f:\SVN\qdos. You may have noticed that I used Unix style directory separators for the repository name – URL – Subversion doesn't really care and as I use Unix most of the time, I'm sort of used to typing them that way around.

Now I have my repository initialised, I need to check out a copy of it to my real working area (on the home PC) so that any work I do there will be under version control. I need to do this because the temporary folder I used to import from is not running under version control, it is simply the location from where I imported the files into the repository. In fact, I need not have copied the files from my USB stick in the first place, I could have simply imported them from the USB stick directly.

Back in my DOS window again:

```
C:\stuff> cd ..\source\qdos
C:\source\qdos> svn checkout file:///f:/SVN/qdos ./
A   part_001.txt
A   part_002.txt
A   part_003.txt
...
Checked out revision 1.
```

Now, I can get rid of the c:\stuff folder that I temporarily saved my work in – it is no longer needed because I have imported it into the repository and I know that it worked because I have checked out a working copy into my c:\source\qdos folder. The same directory tree structure that I have in c:\stuff has now been completely recreated under c:\source\qdos.

When I want to spend a lunch hour working on my code, I plug in the USB drive that my repository lives on and go through the above process to create a working folder – c:\source\qdos and run the svn checkout command again. Because the drive is no longer F: but is J:, I need to do the following set of commands instead:

```
C:\source\qdos> svn checkout file:///j:/SVN/qdos ./
A   part_001.txt
A   part_002.txt
A   part_003.txt
...
Checked out revision 1.
```

Now, I have a working folder at home and one at work. Both talk to the same repository and both work, regardless of the fact that the repository is on F: or J: depending on whichever machine I'm working on. The way that the various Subversion commands 'know' where the repository is because the checkout process creates a hidden directory structure and some files that tell it where to find the repository – amongst other things.

When the time comes to write another exciting episode in the assembly language series, I start typing characters at random into a new file and eventually, save it to the working folder – c:\source\qdos. That does not automatically add the file to version control – a folder under version control can contain 'versioned' and 'unversioned' files and folders.

To make sure that the new file is versioned, I proceed as follows:

```
C:\source\qdos> svn add part_020.txt
A   part_020.txt
```

If I wanted to add more files, I simply add their names to the command line after the first one. I can also use wildcards and so on.

Subversion assumes that all the files and folders mentioned in an import or add command are a single transaction. If it fails to import or add one file out of the list of files/folders, then it rolls back the transaction. It has to be all or nothing whenever you issue a Subversion command.

Continuing with my example, the file is added, but must be committed with a message as follows. I use the status command to find out what has been changed, deleted or added and thus needs to be committed.

```
C:\source\qdos> svn status
A   part_020.txt
```

I can see that part_020.txt has been added and needs to be committed. When I commit a file, a message must be supplied:

```
C:\source\qdos> svn commit -m "Added new file - part 20"
A   part_020.txt
Adding      part_020.txt
Transmitting file data ..
Committed revision 2.
```

Resync when I get home

So, having added and committed a new file, or made some changes to existing files, while at work, I remove the USB drive and go home. Once there, and when I'm allowed 'play time' by my wife Alison, I plug in the USB drive (as F: remember) and issue the following commands from within my c:\source\qdos folder:

```
C:\source\qdos\> svn update
A    part_020.txt
Updated to revision 2.
```

Now, my home folder has been synchronised to my work folder and both have the same files in them – the repository and the Subversion code takes care of any changes, deletions, additions and so forth. When I am finished making changes and have committed them back to the repository, I issue an 'svn update' command the next time I'm at work and I get my home changes merged into my work changes. Easy and no more screw ups on my part – hopefully!

Subversion has lots of other features, but as this is a gentle introduction, I shall miss most of them out. It can help resolve problems – know as conflicts – where I have updated the same area of the same file in both the work and home folders. It cannot do it automatically, but it helps.

If I edit the same file at home and at work, but in completely different areas, it doesn't flag up a conflict and simply merges the two files into one containing both sets of changes.

Now, should I wish to, I can also plug my USB drive into my Linux laptop where it becomes /media/mobile and I can see my repository as /media/mobile/svn/qdos and, I can checkout a copy of my articles and code into a directory on my Linux system.

Subversion is useful for people like me who may have to do some work on Linux and some on Windows. The QDOS/SMSQ systems are like Linux as they have a single CHR\$(10) as the line end character whereas Windows files have CHR\$(13) + CHR\$(10) as the line end. This causes problems but Subversion has a fix. In the Linux working directory, I can tell Subversion to fetch the files in Linux/QDOS/SMSQ format while on the Windows PCs, I can say that I want the files in Windows format – or not as the case may be!

Using with QPC

Using QPC as the 'hardware' of choice is becoming quite popular – and well done Marcel for a splendid piece of software – so many people may be interested in using Subversion to version control their daily boot files, program sources and so on.

It does involve saving files to a DOS drive rather than (or in addition to) your WIN drive.

If you have a folder on your PC mapped to a DOS drive in QPC, then you can set up a Subversion repository somewhere else, import all your files into Subversion then rename the original folder and create a new one with the same name. A quick checkout and you now have a working folder for QPC which is under version control.

If you regularly save files to the DOS device, each time that you exit from QPC you can issue an 'San commit -m "whatever"' command and have all changes submitted to the repository. If you wish to commit only a few files, that is easy too – simply specify their names on the command line.

If you like to use a GUI when running Windows on your PC, then get hold of TortoiseSVN and once installed, it allows you to do much (if not all) Subversion work from the Windows Explorer right-click menu. Tortoise is available from <http://tortoisesvn.tigris.org> – highly recommended.

So, while the subject of Version Control is off topic, I hope you don't mind my little explanation of how I use it for my Articles and code.

Links

Subversion: <http://Subversion.tigris.org>

Free Subversion Book: <http://svnbook.red-bean.com/>

TortoiseSVN: <http://tortoisesvn.tigris.org>

RWAP Services NEWS!

**** We have moved ****

See our updated address details below.

We have also acquired more brand new Sinclair QL membranes and another stock of Epson Stylus Colour 850 inkjet printers, so if you need a better printer for your QL, give us a shout.

More news is always available on our website: www.rwapsoftware.co.uk

We are also looking to produce some new hard disk interfaces for the ZX Spectrum and have a few little projects on the drawing board.

Our websites:

<http://www.rwapservices.co.uk> (General site)

<http://www.rwapsoftware.co.uk> (Sinclair computer second hand and new items)

<http://www.rwapadventures.com> (Adventure Programs)

<http://www.internetbusinessangels.com> (Guidance on setting up online businesses).

New Products!

QWORD 1.0

**NOW WITH DIGITAL
SOUND ON QPC2!**

The wait is now over! Q-Word version 1 is finally available!

Platforms:

QPC/QXL, Q40/Q60, Aurora (with SGC)

Prices:

All versions without P-Word	£ 20.00
All versions with P-Word	£ 30.00

Notes:

Q-Word **DOES NOT** require SMSQ/E with GD2 support -OR- SMSQ/E at all on the Aurora or Qx0 machines. It works on the highest colour depth everywhere regardless of Operating System.

The Aurora version is available on either HD or ED disk. For the latter add £1.00 to the price. ED version is uncompressed and can be run directly from the floppy. All other Floppy versions are compressed. QPC/QXL version comes on CD. Non CD versions DO NOW support digital sound on QPC2



for **Windows**

For QLers that run Windows or with incompatible hardware for Talent Games, we now have re-released these adventures so that they can run on your Windows-equipped PC. No Emulator, floppies, microdrive backups etc. required, just a one-click install! Of course the full QL line is still available! (See side column)

Talent Games for Windows ea. £ 10.00
(Each Game includes a runtime installation of QLAY-2 by Jimmy Montesinos)

Games Currently Available from www.rwapadventures.com

The Lost Kingdom of Zkul
West
The Prawn
Return to Eden

Replacement Sinclair QL Keyboard Membranes

We always have a stock of brand new Keyboard Membranes (and keyboard parts) for the original Sinclair QL, so if you have some keys which no longer work, just give us a call.

Cost is only £18.50 plus £2.75 post and packing.

Second Hand Items - Huge Range Available

We stock a wide range of books, hardware and software for the Sinclair QL, Z88 and ZX Spectrum, including disk interfaces, memory expansion and microdrive cartridges. If there is anything you need - have a look at www.rwapsoftware.co.uk (or ring us with details of your requirements).

We are always happy to help.

RWAP Services

3 Dale View Court, Fulford, Stoke-On-Trent, Staffordshire ST11 9BA TEL: (+44) 1782 398143 From the UK Dial: 01782 398143

Website: <http://www.rwapsoftware.co.uk>

Email: sales@rwapsoftware.co.uk

We Accept Payment using:



(Cheques in £ sterling made payable to R. Mellor)

Old Favourites!

Utilities

SBASIC / SuperBASIC Reference Manual on CD	£ 20.00
Sidewriter v1.08	£ 10.00
Landscape Printing (EPSON printers)	
ImageD v1.03	£ 10.00
3D object generator	
Q-Help v1.06	£ 10.00
Superbasic On-Screen help system	
Q-Index v1.05	£ 5.00
Keyword-to-topic finder	
ProForma ESC/P2 Drivers v1.04 for ProWeSs Printer Driver	£ 8.00

Applications

Flashback SE v2.03 (upgrade only)	£ 5.00
Database	
QL Cash Trader v3.7	£ 5.00
Accounting/Finance	
QL Payroll v3.5	£ 5.00
Accounting/Finance	
QL Genealogist v3.26	£ 20.00
Genealogy	
Genealogy for Windows	£ 50.00
QL Genealogist to Windows version upgrade	£ 25.00
QL Cosmos v2.04	£ 5.00
Planetarium	
Q-Route v2.00	£ 25.00
Route Finding	
Upgrade from v1.xx	£ 5.00
Britain map v1.11	£ 2.00
BIG Britain map (needs 2Mb) v2.03	£ 5.00
Various Britain Area maps (ask for details)	ea. £ 2.00
Ireland map v1.00	£ 5.00
Belgium map v1.01	£ 2.00
Catalonia map v1.03	£ 2.00
P-Word UK English Dictionary (500.000 words!) Dictionary	£ 15.00

Leisure

Return to Eden v3.08	£ 10.00
Adventure	
Nemesis MkII v2.03	£ 8.00
Adventure	
The Prawn v2.01	£ 8.00
Adventure	
HorrorDay v3.1	£ 8.00
Adventure	
West v2.00	£ 5.00
Adventure	
The Lost Kingdom of Zkul v2.01	£ 5.00
Adventure	
All 6 games above	£ 25.00
D-Day MkII v3.04	£ 10.00
Strategy/War Simulation	
Grey Wolf v1.08	£ 8.00
Graphical Submarine Simulation	
War in the East MkII v1.24 (upgrade only)	£ 5.00
Strategy/War Simulation	
Open Golf v5.20	£ 8.00
Sports Simulation	
QuizMaster II v2.07	£ 5.00
Quiz	
Stone Raider II v2.00	£ 5.00
Arcade Game	
Hoverzone v1.2	£ 5.00
Arcade Game	
Deathstrike v1.5	£ 5.00
Arcade Game	
Flightdeck v1.0	£ 10.00
Flight Simulation	
All 6 games above (Open Golf, QuizMaster II, Stone Raider II, Hoverzone, Deathstrike and Flightdeck)	£ 28.00

Notes on Software requirements

The following programs have a minimum SGC card requirement: P-Word, Qword, Big Britain MAP for Q-Route

Minimalist Games - Part 2: Horses

by Stephen Poole

Recently I re-read an old book on ZX programs, and tried to adapt one of their horse-racing games for the QL. Unfortunately, their 25 lines of condensed code does not easily translate onto the QL, so I decided to rewrite everything from scratch.

The result is this 'Horses_bas' program, which, as you can see, only occupies 4 lines of code, line 150 and also lines 210 to 230. All the rest serve just to make the initialised QL conform to the ZX defaults. In other words, the game uses but eight SuperBasic statements. So I just had to include it in the 'Minimalist' series.

The ZX listing used four nested FOR-loops (twice), set up two 2-D arrays, and of course used their famous 'N=NOT P1' to use less memory rather than 'N=0', and 'N=SGN P1' in the place of 'N=1' etc, the whole lot peppered with ZX specifics such as: PRINT 'n' AND n=1 : !! My first attempt at coding this for the QL used

SElect structures to manage the moving horses, (which simplified indexing), and which then provided a neat insight as to how to reduce these SElections into a one-dimensional array of counters.

The advantage of this game is that all the family can play it at the same time, and it is compulsive because you just have to keep trying until you win. But as in similar cases, neither QL Today nor I can accept any responsibility for any misuse you may make of it. I hope that young members of readers families will be interested in the program, as these days there are very few books or magazines that could inspire young readers to start programming for themselves. Another good reason for reading Quanta and QL Today! To improve the Randomness, use Mark Knight's True_Randomise function. (Now is the time to revise back-issues of QL Today..).

```
100 ::
110 REMark Horses_bas, by S.Poole, v14jun2005
120 CLEAR: OPEN#1,con_16: WINDOW 512,256,0,0
130 PAPER 0: STRIP 2: INK 7: CLS: RANDOMISE DATE
140 :
150 n=RND(3 TO 9): DIM ct(n): cols=83
160 FOR j=1 TO n: AT j,0: PRINT j
170 AT 19,1: PRINT 'Choose a horse...then HIT its key to race';
180 h$=INKEY$(#1,-1): AT 19,1: CLS 3
190 :
200 REPEAT race
210 h$=INKEY$(#1,3): hs=RND(1 TO n)
220 AT hs,ct(hs): PRINT ' ';hs: ct(hs)=ct(hs)+1
230 IF ct(hs)=cols: EXIT race
240 END REPEAT race
250 :
260 AT 21,9: PRINT 'Horse!'hs!'Won!!' , 'Another Game? y/n >'
270 IF INKEY$(#1,-1)=='n': STOP: ELSE : GO TO 100
280 ::
```

My Holiday Romance

by David Denham

It was hard. It was really hard. But I had made a decision to go for change and I had to stick to it. Since retirement, I've travelled quite a lot and despite my love for my trusty QL and Gold Card, it was becoming more and more obvious to me that travelling around in a camper van was not really compatible with my QL system. Indeed, it was asking for trouble.

I mean, lugging around a QL monitor, a QL, a Gold Card, a set of disc drives, printer and interface, not to mention power supply and all sorts of other cables did not really make for a restful domestic environment (even though she often pinched time on it to use Quill to write something while we were away), not to mention available

space. Never mind the day when I pulled over in a services area and found all the QL gear on the floor in the back having made a rather poor attempt at breaking out and starting a life of its own as I had earlier swerved to avoid a pedestrian in a position no pedestrian should be in.

Scratches and grazes notwithstanding, the equipment survived that particular incident. My nerves didn't. So in an effort to prove how much I loved my QL I took the decision that it was high time my wife and I went on our travels and the QL stayed at home.

Thus began the affair with my laptop PC. My wife had been keen to buy one anyway, so in the interests of domestic harmony (roughly translated means "for the sake of my QL") we purchased a used laptop for a good price - if we didn't get on with it we wouldn't have lost much money on it - so that she could have a mobile word processor and I could have a QL emulator running on it.

Not being sure how successful this would be, I decided I wouldn't buy a QPC or even a Qemulator. There is a free option - QLay.

QLay in fact exists in three main incarnations. The original was written by Dutch author Jan Venema in the 1990s, and was produced for DOS, Windows and Linux platforms. Since then, there's been a QLay2, which is an update of the original QLay for Windows. That work was done by Jimmy Montesinos who lives in France. There is a program called QL2K also from Mr Montesinos, but you need to register to use it (I'm not sure if it's free or not), so I started off with QLay 2 and intend to stay with that for now, at least.

The thought of mastering Windows plus mastering an emulator was attractive as a challenge, but not if I wanted to get this up and running before the next trip so that my QL could spend the holiday at home and I could indulge in what might prove to be an exciting little holiday romance with my new computer.

Actually, Windows was already familiar to me to some extent, the main worry was what I would do if I couldn't get the emulator to work. I've used Windows programs plenty of times, but if the emulator didn't work first time, I was not confident of my ability to sort it all out. Could I survive two weeks away from home without a QL? Would I end up for two weeks with only Windows for company? Or would the whole lot be consigned to the bottom drawer (the Denham equivalent of a certain ex-editor's renowned threats to chuck his PC out of windows with a small W). Well, I thought, if it all goes wrong I'll murder the PC, just do the time and come out two weeks later a reformed man.

In practice, it all went fairly smoothly with but a few minor hiccups which were soon overcome, and this is the tale of David Denham's introduction to QL emulators and how he came out the other end safely and without a PC-induced nervous breakdown.

The first step was to get hold of a copy of QLay 2. This was downloaded from Jimmy Montesinos's website at

<http://www.jadnam.org/QL/QLAY2/>

You can download a minimal set of files just to use the emulator (QLAY2.ZIP), or you can download a complete package including source files (QLAY2-Full.ZIP). I chose the former. Anyone who wants to study the sources is encouraged to do so and contribute to its development.

It is well worth making a printout of the instructions on the site. They were apparently written by Simon Goodwin, known to QLers as the author of Speedscreen and Supercharge compiler, and someone who has written extensively on the subject of the QL over the years, including his extensive DIY Toolkit files collection. These notes on QLAY2 include both instructions and a lot of background information. They supplement the instructions which come with the QLay2 package quite well. QLay2's documentation is a bit patchy in fact, many documents based on the original QLay ones, with updates here and there where relevant. While the information is all there, digesting it all can be quite a challenge. For example, the main body of the text of the README.TXT file (the main emulator instructions) consists of Jan Venema's original QLay documents, interspersed with Jimmy's update notes flagged by preceding > marks, like email quotes, which does not make it very readable, although all the information you need is there.

Once downloaded, simply unzip the package to a directory of your choice on your Windows hard disc and create a shortcut to the QLAY2.EXE file, which you will use to start the emulator in future. The emulator comes with two ROM images, namely a copy of Minerva 1.98 and a ROM image called NFA.ROM (Native File Access - used to store QL files on the Windows hard disc in QLay's own format - more about this later).

You don't have to use Minerva - QLay will happily use any QL ROM image such as a JS or JM ROM. It can also be handy to get hold of a copy of a Toolkit 2 ROM image. I believe it's now freely available from Dilwyn Jones's website, or if you have a QL with an external plug-in Toolkit 2

EPROM, you can save a copy from there with the command

```
SBYTES f1p1_tk2_rom,48*1024,16*1024
```

(i.e. the ROM slot appears 48KB into the memory map and the EPROM is 16KB long).

The emulator has three ROM slots, so you can use a QL ROM image in the BOOT slot, a Toolkit 2 ROM in the next slot and the NFA.ROM in the third slot.

Once you have installed the ROM images in the same directory as the QLAY2.EXE program, you are ready to start the emulator, but there are two other programs you should really install, QLAYT090.ZIP and QLTOOLSQ.ZIP. These are tools programs to aid with the transfer of files between the emulator, floppy discs and the Windows hard drive. At this point you come to realise what is possibly QLAY's main weakness, its lack of floppy disc handling. You can't load and save from and to a QL floppy disc. What you have to do is to use the tools programs to transfer files between the emulator's directories on the PC's hard disc and the floppy discs. Even then, life is not that simple, as the QLAY directories are essentially Windows hard directories, but with the QL file headers stored in a special directory file called QLAY.DIR. When you save something in the emulator, you have up to 8 WIN drives and another 8 MDV files. MDV files are essentially a directory on the hard disc which pretends to be a microdrive. I never really got to grips with the QLAY2 MDV drives, and the WIN drives were more convenient to use for most purposes. But you have to be aware when using the WIN drives that saving or transferring files makes proper use of the QLAY.DIR files or things may not go to plan - if a QL file does not have its header stored in QLAY.DIR, QLAY won't know it exists and any QL executable job header may be lost. Unless QLAY has an entry for a file in its own directory, the file may exist where it ought to be, but will be ignored by QLAY and it will to all intents and purposes be nothing more than a simple Windows data file. And we all know that PCs don't understand QL file headers - they simply get lost and the program can no longer be executed.

This is where the QLAY tools programs come in and they are essential if you intend to transfer files into QLAY or out of QLAY (e.g. to or from QL floppy disc). What the tools programs do is copy the file in the direction indicated, but also try to make sure that the QL file headers are copied correctly too so that QLAY knows about them, and so that the QL will know about them when the disc is read on a normal QL.

QLTOOLSQ v2.7 is a special version of the original QLTOOLS software which was designed for QL/DOS file transfer. QLTOOLSQ is a version modified for use with QLAY, which can copy all files from a QL format disc in the PC's disc drive. QLAYT is more about moving individual files or lists of files into or out of QLAY's directories (normally transfers between Windows and QLAY). It can be quite a clumsy affair when you need to send the output of a DOS DIR command to a file, and use that file as a list of files to transfer. Options exist to specify dataspace and so on, and it can take a while to get used to all the available options. It is worth setting up qltoolsq and qlayt though, as they are the only really effective way of moving files around. What you do is to go into a DOS box in Windows (usually either find Command Prompt in the start menu, or using the RUN command in the Windows start menu, enter RUN CMD to get the command prompt going. Then you type in the name of the program along with whatever parameters the commands need to perform the action required. Let's take a brief break here and study what QLAY 2 has to offer.

QLay emulates the hardware of a Sinclair QL with 128K, 640K or up to 8MB of RAM. It works with most QL ROM images. As it uses QDOS, it can only offer 512x256 screen resolution since QDOS doesn't know any better, and QLAY can't run SMSQ/E. Screen modes 4 and 8 are both available, together with the dual screen capability of the QL, although unsupported by standard QDOS unless you have a Minerva able to handle the second screen. QLAY can cope with pointer environment. Keyboard emulation via IPC (the QL second processor). A small list of international keyboard variants (British, American, German, French and Italian). A mouse is emulated via memory mapped I/O, allowing the PC's mouse to be used with QLAY if pointer environment is used. The original QLAY had IPC sound emulation, but since QLAY2 is aimed at more recent versions of Windows such as NT, Windows 2000 and XP, sound is not available in QLAY 2 due, the author says, to limitations of NT Kernel Architecture. QLAY 2 handles up to 8MB RAM and has a full 68000 emulation, not just 68008 as on the original QL, including all exceptions and trace. QLAY 2 needs a minimum of a Pentium CPU as opposed to a lesser requirement of a 386 CPU for the original QLAY. 9MB of RAM is needed to run the emulator, as opposed to 4MB for the original QLAY, and a minimum of 8MB of hard disc space (4MB if Windows swap files used).

The manual lists some command line switch options for starting QLAY, but these seem not to be implemented in QLAY2, instead these are set within a QLAY2.INI file.

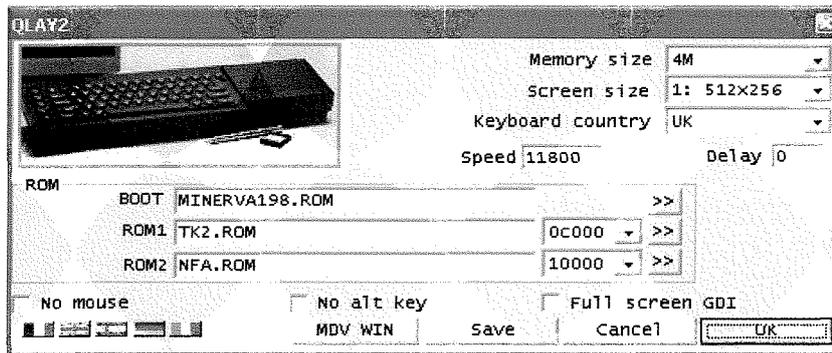


Figure 1 - The startup configuration screen

There's quite a lot of options here, but once set you probably won't need to change it again. The list of flags across the bottom selects the language used for the startup screen - just click on the flag for the language required. It defaults to French, but English, Spanish, Italian and German are also available. As you click on the flag, the language displayed in the menus changes immediately.

From the top, the first list lets you select the memory size of the emulator. The usual choice is 1MB, which is fine for most things. You can select values from 128K like an unexpanded QL (useful for very old games which only run on standard 128K systems) right up to a maximum of 8MB, which is twice the memory of even a Super Gold card!

The next box gives the impression of selecting screen resolution, but in fact this is not quite so. Normally, you will select 512x256 like a standard QL, but this occupies only a tiny part of the screen on a modern PC, so QLAY2 rethinks the whole thing and lets you choose to use more than one PC pixel per QL pixel, so that the display is still 512x256 as far as the QL is concerned, but the PC uses more pixels to magnify the picture in effect, making the display look bigger. There's a total of eight sizes to choose from, use whichever suits your requirements and eyesight best.

Next down is the keyboard country required. This lets you specify US, UK, German, French or Italian keyboard layouts. The inclusion of US is especially fortunate for me as my laptop's keyboard seems to conform to US type, I don't know if this is the norm or not.

The next boxes control the speed and delay of the emulator. Speed is especially critical.

The Speed item lets you fine tune the emulator's internal timings. It's not a very easy thing to explain, as it suggests something like how fast the emulator runs, but it's not that easy. Basically, it helps you to set the emulator so that timing-critical things like keyboard repeats and PAUSE delays are correct, so that they correspond to a real QL. Get it wrong, and you may find that the keyboard response becomes way too fast (press a key and you get half a dozen characters instead of the one you expected), or PAUSEs do not wait for the time you expected

them to wait. This is how the instructions describe it:

"...it sets the ratio of the emulated QL speed to that of an original QL so that processor-timed operations like key repeat are user friendly on all of the wide range of Windows hosts. The SuperBASIC date and time functions are not affected, as QLAY uses the PC's clock for those." As the manual suggests, watch the flashing cursor and you can often tell if it's not set right as the cursor will flash faster or slower than a QL. Fortunately, there is a simple way to set the speed factor, all you need to do is make a note of the existing Speed value from the setup screen as the emulator starts, then run a small SuperBASIC program called FT_bas for one minute. This program is supplied with the emulator. It will tell you if the speed setting is not right for that particular PC and tell you by how much you need to multiple or divide the existing speed value.

For most PCs, the speed value is usually somewhere in the range of 500 to 1200. The faster the PC, the higher the value. After it has run for one minute, the program tells you how much to change the value. Go back to the startup screen and enter the revised value in the SPEED box. Then click on SAVE to make sure it remembers this new value every time it starts up.

For example, if the existing SPEED value is 500, and it tells you to multiply this value by 1.5, you would enter a value of 750 for SPEED.

FT_bas is a very short BASIC program which simply does 60 one second PAUSEs and compares this with the "real" one minute as determined by the QL clock, which is synchronised to the PC one which is known to be accurate, of course. From this, it can tell how many seconds per minute that the existing SPEED value loses or gains and calculates how much it needs to be changed to correct that difference.

The DELAY value seeks to slow down the emulator to cope with games that run too fast. It doesn't always work, as some older games derive their timing directly from 68008 processor activity, which cannot be emulated 100% accurately, as this is really a PC processor interpreting machine code for another completely different processor. The delay setting is found largely by trial and error - in my case a value of 0 or 1 usually works well, but may vary from PC to PC. It is a question of setting the right compromise between game speed and keyboard response, for example. Set the SPEED value first with FT_bas before you try to set the DELAY value.

One thing you will find when running QLAY2 is that no matter what speed and delay values you set, the load on the PC processor is pretty constant. Minimise QLAY 2 and run something else, it will usually be slow and sluggish, depending on the PC processor power of course. QLAY must hog a significant percentage of processor time all of the time, but it seems to drop if you click on one of the menus, as though the "QL" is suspended at that point. One way of relieving this load on the PC is to click on the QL menu and select the Exit command and leave the emulator parked on the "Are you sure Yes/No" dialogue, as strange as that sounds.

The next set-up features are the ROM slots. These emulate the ROM addresses on the original QL. The first of the three ROM slots is for the operating system ROM, usually a Minerva or JS ROM, of size 48 KB. The second corresponds to the back EPROM slot on a QL, and happily takes the Tool kit 2 16K ROM image. The third slot occupies address space just above the back EPROM slot's 16K address range, at address 65536 in decimal, which I think is used for hardware I/O etc. on a real QL. This third slot usually takes the NFA.rom which is a piece of code allowing QLAY to use the Windows hard disc to store QL programs, as a WIN file. NFA stands for Native File Access. Note that it does not understand QXL.WIN files at all, although one of the tools programs is supposed to be able to extract files from a QXL.WIN and copy them to the QLAY directories. I haven't tried this.

The NO MOUSE option lets you disable the QLAY mouse if you find that the Windows pointer and the QLAY pointer both show on screen at the same time in different positions. If you use this option, there is no mouse facility when using pointer environment with QLAY, although the cur-

sor keys can be used to drive the pointer instead, of course.

NO ALT KEY makes QLAY ignore one of the ALT keys on a PC keyboard so that it can be used for Windows shortcuts. The other ALT key is always available to the emulator anyway irrespective of this setting.

FULL SCREEN GDI scales the QLAY display to fill the screen. If you find it annoying that the QLAY window only uses a small part of the PC's larger screen display, use this option to allow the QLAY display to expand to fill the screen. Resolution is still 512x256 - this just makes the PC use several Windows pixels per QL pixel to scale it to fit the PC screen. Depending on how good your PC is at scaling graphics and what PC screen resolution you use, it may produce good or very ugly results because you may find that the scaling needs to use, say, 2 pixels for some QL lines and 3 pixels for other QL lines to scale it correctly to fit the whole screen. This means some horizontal lines are thin, some thicker, and it makes for a very ugly result in extreme cases, especially with text screens. Again, largely a question of trial and error to get the best compromise possible. If you can't make the full screen GDI work to your satisfaction, try the various SCREEN SIZE options instead, as these can use consistent numbers of PC pixels to QL pixels to make the display look normal even if they don't quite fill all of the screen.

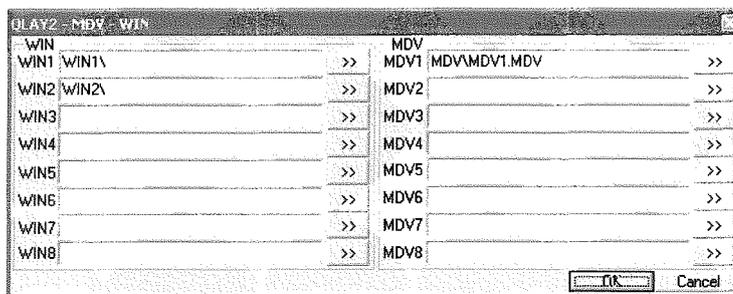


Figure 2 - the mdv/win seupt screen

Finally, we come to the MDV/WIN button. Click on this and a complex looking screen appears which lets you define where QLAY2's WIN and MDV drives live on the PC hard disc. Think of the QLAY WIN drives as being up to 8 separate hard discs as far as QLAY is concerned. Each can store up to 159 files, giving you over 1200 files in total, more than enough for most people I'd have thought. The MDV drives are very small (255 sectors, which is the maximum theoretical capacity of a microdrive cartridge, though most cartridges are about 200 to 220 sectors) drives, intended to

emulate microdrives on the original QL. It can be useful to transfer older games into one of these to see if you can get them to work on QLayer MDV if you can't work out how to configure the game to run from a WIN drive.

Although it can only handle 8 drives at a time, you can set up as many as you like and switch between them using the setup screen if need be. So you could set up one copy of WIN1_ to contain your work information, another for your home use, another for your partner and so on.

Hard discs for a QL are a subject new to me, though I've made use of directories and so on on a floppy disc drive with a Gold Card. But QLayer doesn't use directories - it only has what is called a Level 1 filing system, no MAKE_DIR command, so you can't go putting all your programs into separate directories. So think of each WIN drive as a large capacity floppy disc drive called WIN1_, WIN2_ and so on. In a way, the limit of 160 or so files per WIN drive is a good thing. Imagine trying to find your files amid thousands of other files!

drives in the same PC directory as QLayer itself. So, I put QLayer2 into a folder called QLayer on the C: drive. I created folders called WIN1, WIN2, WIN3, WIN4 and so on inside the QLayer directory from Windows, then used the browse buttons in the MDV-WIN screen (the » buttons to the right of each drive name) to find these folders and point QLayer to them. Although I did set up 8 MDV drives as well, I found them less useful and haven't really used them since then. Once all the



Figure 4 - Minerva starting up

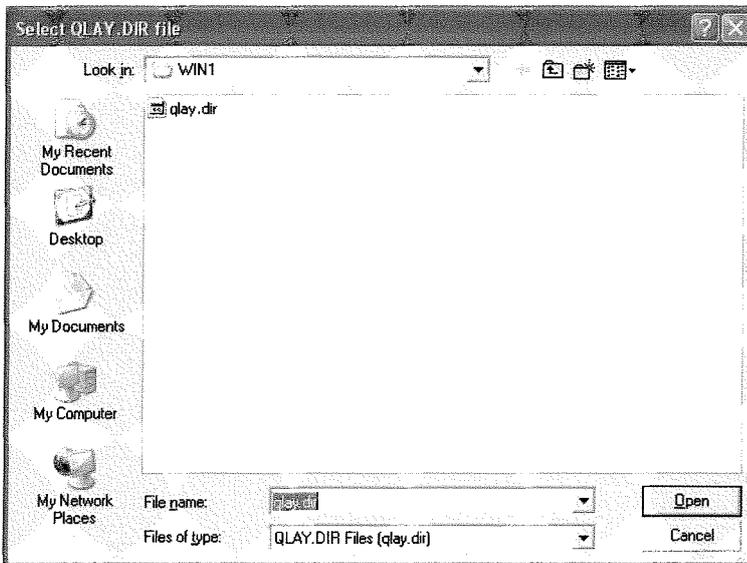


Figure 3 - browsing to find a QLayer WIN drive

This actually finds folders with a QLayer.DIR file, a file which contains the headers for the QL files stored, so it does manage to give the impression of having to create something which already exists, but don't be too worried about it, it's quite easy. If you get stuck, you can always go into Windows and copy a QLayer.DIR file from one place to another to give you a head start. I found it most convenient to put these WIN

drives are set up, click on the OK button to return to the main setup screen, then click on the SAVE button to store these settings for future use. Finally click on the OK button in the main setup screen and the emulator starts.

The familiar QL screen is shown and it prompts you to press F1 or F2 for Monitor or TV windows mode as a standard QL would. If using the Minerva ROM, the emulator may get fed up of waiting for you after a while and start up of its own accord. This was a feature built into Minerva in the days when bulletin boards rather than the internet

were the norm, so when a power cut occurred, or the QL reset for some reason, the QL could start itself up automatically and run its boot program to restore normal service again when the power was restored.

OK, we now have an emulator set up and ready to go and up to 8 drives with nothing in them, so where do we go from here?

If you haven't already done so, I run the FT_bas program to make sure the emulator speed is correct.

he next stage will be to transfer some files from the real QL using a QDOS floppy disc.

To do this, we need the qltoolsq program supplied with QLayer2. This is a DOS program which we run from Windows. Make sure that the folder which contains QLayer2's WIN drive contains copies of the PC programs called QLAYERTEXE and QLTOOLS.EXE.

QLAYERTEXE helps with transferring files between Windows and QLayer2.

QLTOOLS.EXE helps with handling QDOS formatted floppy discs and transferring files between the QL floppy disc and the QLayer WIN drives.

The simplest way of running these is to start a DOS command box in Windows. In the Windows START menu, go to the Accessories sub-menu and click on Command Prompt (not sure if this varies between different versions of Windows). Alternatively, use the Run command from the START menu to run "cmd" (without the quotes). We need to change to the QLayer2 directory:

```
cd qlay\win1\
```

ensure there is a QL floppy disc in the PC's disc drive, with some files on it. Now enter the command:

```
qltools a: -d
```

This will list the files on the QL floppy disc. To transfer these files to QLayer and update QLayer's directory (contained in a file called QLAYERDIR) enter this command:

```
qltools a: -q
```

Qltools has many other commands. This one is probably the most useful, as it transfers all files from the disc in one go, and creates the entries in the QLayer2 directory automatically. Copying files from Windows or from the QLayer directories to the QL floppy disc is also possible, using "qltools -w filename" (without the quotes) which copies the file specified to the floppy disc.

Qltools.exe has its own instruction files, but you can get it to display a help screen by simply starting it with nothing after the qltools command.

QLayer.exe is a program to copy, update, delete etc files between Windows and QLayer2. Using something like Windows Explorer it is easy to copy or drag and drop files into the folder which

is the QLayer2 WIN file. But just copy it there and QLayer does not see it, because it hasn't yet been added to the directory list in QLAYERDIR. This is where QLAYERTEXE comes in.

There are two ways of using this DOS program. The first is to act directly on files, the other is to create a list of files and let qlayer work on the list instead.

If your QLayer WIN drives do not contain a file called QLAYERDIR, use QLAYERTEXE to create one while adding files.

Firstly, send a list of files to be transferred to a PC file called list.txt - we can use the DOS command DIR with a '>' command to send the output to a file instead of to the screen:

```
DIR /b >list.txt
```

The /b forces the DOS command DIR to just list the simple filenames without details of length etc to make the list easier to handle.

Then we use the following command to create a new QLAYERDIR and add the files listed in "list.txt" to QLayer's directory:

```
qlayer -c list.txt
```

The '-c' forces qlayer to create a new directory file if none exists already. If you just want to add (append) the files list to an existing QLayer directory without creating a new one, use -a instead:

```
qlayer -a list.txt
```

After creating or adding files to QLayer's directory, you may feel more comfortable checking it before you proceed. To do this, use qlayer -l (that is, a lower case letter L not the number 1):

```
qlayer -l
```

This will show if the file is listed in QLAYERDIR, which will ensure QLayer2 sees the file. If you wish to, you can even send this list to a file, e.g. if you are a neat and organised person who likes to keep accurate records of files, or simply to create a list to help you with copying files from the emulator to Windows:

```
qlayer -l >qlaylist.txt
```

Using lists like this is a bit clumsy but OK for a long list of files because you don't have to type in a qlayer command for each and every file. But if you want to add just one file from Windows to QLayer, it is easier to use the 'include' command option to specify that you wish to append just one file to QLayer's directory. In Windows, copy that file to the folder holding QLayer's directory then tell qlayer to inform QLayer that the file is now

TF Services

Compswitch

A UK 4-way trailing socket designed to switch off computer peripherals automatically when the computer is switched off, or (in the case of an ATX computer) when it auto-powers down. *Compswitch* has one control socket, and three switched sockets. Can be used with lights/hifi/monitors—ie a QL monitor can be used as a switch control.

Cost **£24**

superHermes

A major hardware upgrade for the QL
All Hermes features (working ser1/2 at 19200, independent baud rates/de-bounced keyboard/keyclick) IBM AT kbd I/F // HIGH SPEED RS232 at 57600// serial mouse port and 2 other RS232 inputs// 3 I/O lines // EEPROM

Cost (including manual/software) **£90** (£92/£93)
IBM AT UK layout Keyboard..... **£11** (£13/£15)
Serial mouse..... **£8** (£8.50/£9)
Capslock/scrolllock LED **£1** (£1.50/£1.50)
Keyboard or mouse lead **£3** (£3.50/£3.50)
High speed serial (ser3) lead..... **£4** (£4.50/£4.50)

Hermes available for **£25** (£26/£27) Working ser1/2 and independent input, debounced keyboard.

SuperHermes LITE: All Hermes features (see above) + an IBM AT keyboard interface only.

Cost (incl keyboard lead) **£53** (£54/£55)

QL REPAIRS (UK only)

Fixed price for unmodified QLs, excl microdrives. QLs tested with Thorn-EMI rig and ROM software.

£27 incl 6 month guarantee

Minerva

The ORIGINAL system operating system upgrade

OTHER FEATURES COMMON TO ALL VERSIONS

DEBUGGED operating system/ autoboot on reset of power failure/ Multiple Basic/ faster scheduler- graphics (within 10% of lightning) - string handling/ WHEN ERROR/ 2nd screen/ TRACE/ non-English keyboard drivers/ "warm" fast reset. V1.97 with split OUTPUT baud rates (+ Hermes) & built in Multibasic.

First upgrade free. Otherwise send **£3** (+**£5** for manual if reqd).
Send disk plus SAE or two IRCs

MK1.. **£40** (£41/£43) MK11.. **£65** (£66/£67)

MINERVA RTC (MKII) + battery for 2.56 bytes ram.
CRASHPROOF clock & I²C bus for interfacing. Can autoboot from battery backed ram. Quick start-up.

QL RomDisq

Up to 8 mbyte of flash memory for the QL

A small plug in circuit for the QL's ROM port (or Aurora) giving 2, 4 or 8 mbytes of permanent storage - it can be thought of as a portable hard disk on a card, and reads at some 2 mbytes per second.

Think of it - you could fully boot an expanded QL, including all drivers/SMSQ etc off RomDisq at hard disk speed with only a memory expansion needed.

2 mbytes RomDisq..... **£39** (£40/£41)
4mbytes RomDisq..... **£65** (£66/£67)
8 mbytes RomDisq..... **£98** (£99/£100)
Aurora adaptor..... **£3** (£3.50/£4)

MPLANE

A low profile powered backplane with ROM port

A three expansion backplane with ROM port included for RomDisq etc. Aurora can be fitted in notebook case and powered off single 5V rail - contact QBranch for details. Two boards (eg Aurora and Gold Card/Super Gold Card/Goldfire fixed to base. Suitable for Aurora (ROM accessible from outside) & QL motherboard in tower case. Specify ROM facing IN towards boards, or OUT towards back of case.

Cost **£34** (£35/£36)

I2C INTERFACES

Connects to Minerva MKII and any Philips I²C bus

Power Driver Interface 16 I/O lines with 12 of these used to control 8 current carrying outputs (source and sink capable)

2 amp (for 8 relays, small motors)..... **£40** (£43/£44)
4 amp total (for motors etc)..... **£45** (£48/£50)

Relays (8 3a 12v 2-way mains relays (needs 2a power driver)..... **£25** (£28/£29)

Parallel Interface Gives 16 input/output lines. Can be used wherever logic signals are required..... **£25** (£27/£28)

Analogue Interface Gives eight 8 bit analogue to digital inputs (ADC) and two 8 bit digital to analogue outputs (DAC). Used for temp measurements, sound sampling (to 5 KHz), x/y plotting..... **£30** (£31/£32)

Temp probe (-40°C to +125°C)..... **£10** (£10.50/£11)

Connector for four temp probes..... **£10** (£10.50/£11)

Data sheets..... **£2** (£2.50/£3)

Control software & manual (for all I/F)..... **£2** (£2.50/£3)

QL SPARES

Keyboard membrane **no longer on sale**

1377 PAL **£3** (£3.50/£4)

Circuit diagrams **£3** (£3.50/£4)

68008 cpu or 8049 IPC..... **£8** (£8.50/£9)

8301/8302 or JM ROM or serial lead..... **£10** (£10.50/£11)

Power supply (sea mail overseas)..... **£12** (£19/£23)

Other components (sockets etc) also available

Prices include postage and packing (Airmail where applicable) Prices are: UK (Europe /Rest of world). Payment by cheque drawn on bank with UK address/postal order or CASH! I can no longer accept card payments as UK only does PDQ transaction. SAE or IRC for full list and details

22 Feb 05

29 Longfield Road, TRING, Herts, HP23 4DG

Tel: +44 (0) 1442-828254

Fax/BBS: +44 (0) 1442-828255

tony@firshman.co.uk

http://www.firshman.co.uk

added to its directory. Suppose it's a file called READMETXT:

```
qlayt -i README.TXT
```

Basically, as QLAYDIR holds the QL file headers (details of length, file type etc) for the files which live in QLAY's WIN folder, all it does is to add the 64 byte file header for READMETXT into QLAYDIR so that QLAY2 becomes aware it exists. QLAY2 should not be running when you use QLTOOLS or QLAYT in this way, it only becomes aware of changes when it is first started.

If the file to be added is a QL executable job, things are a bit more complicated since Windows doesn't understand QL file headers, so as far as QLAY would be concerned, a QL program is just another data file, not a program. Suppose we have a QL program called TEST_TASK which has been copied to QLAY's folder and we wish to make QLAY aware the program exists and is in fact an executable QL program. We have to make sure that QLAY2 has a suitable file header which includes the dataspace value needed by that program when it runs on a QL. Some programs, such as those compiled with C68, contain the dataspace value in something called the "XTcc" field, embedded in the file itself. QLAYT can scan for this and try to extract the dataspace for the program:

```
qlayt -i test_task -X
```

The -X, the letter after the - ("switches") must be in the correct case, tells QLAYT to try to find the dataspace from an XTcc field in the program itself. This doesn't always work, so you may have to resort to plan B - specify the dataspace yourself:

```
qlayt -i test_task -d 2048
```

the "-d 2048" indicates that the directory entry for this file should be for an executable file with a dataspace value of 2048 bytes (it should be a decimal number).

If you are unsure of the dataspace value required for a given program, you can check it on your QL before transferring the file to QLAY. If you have Turbo compiler, use its dataspace_task program to tell you the dataspace, or use the FDAT function of Toolkit 2:

```
PRINT FDAT(\f1p1_test_task)
```

If you are completely unsure, you can often get away with giving a safe, high value. Programs don't mind being given too much dataspace as a rule, although it is wasteful of free memory.

If all else fails, you can transfer the "program" to QLAY as a data file, then use ALCHP/ LBYTES/ SEXEC commands in QLAY2 to convert to an executable. Another method which is harder work is to zip up a program on a QL using the QL ZIP program and transfer the zipped program to QLAY as a data file then use QL Unzip on QLAY to unzip the program. Being held in a QL _zip file protects the job header and dataspace values.

The QLAYT manual lists several ways of transferring files from a QXLWIN to QLAY directories, although I didn't have a QXLWIN to hand to test this. I suppose you need to be aware that QLAY WIN drives can only have a maximum of 159 files each, and that QLAY2 doesn't handle level 2 directories. Use the -E option to Extract files from a QXLWIN in this way. First, create a list, then add the content of the list to QLAY2's directory:

```
qlayt -E qxl.win >list.txt  
qlayt -c list.txt -q qlay.dir
```

The first line extracts a list of files from the QXLWIN. Using a list has the advantage that you can load the list into Windows Notepad editor and manually alter the list if required. For example, if the QXLWIN contains thousands of files you can simply delete the ones not needed to make the list much shorter. The second command creates a new qlay.dir (the -c option) and puts the files listed in list.txt into it, then the -q option ensures that the new directory file created is called QLAYDIR. I should emphasise, I didn't have a QXLWIN to hand and the above example is based on my assumptions and understanding of the manual and I am probably completely wrong (assumption is the mother of all foul-ups, of course!)

A simpler method is probably to extract and add direct with something like:

```
qlayt -E qxl.win -q qlay.dir
```

If you wish to transfer files to a MDV directory instead, use -C (note: upper case C this time!):

```
qlayt -C list.txt
```

This will create an MDV image file from the files listed in list.txt.

You can copy files from an MDV slot to a WIN directory with the -W option:

```
qlayt -W mdvfile
```

Most of these commands use the current QLAYDIR directory. Adding a "-q name" part to the command will tell it to use another directory file instead if required.

In addition to the example commands given above, there are commands to remove filenames from QLayer2 directories, convert files between QDOS and Windows formats and vice versa, update header information if a file in a QLayer2 directory has been altered by Windows and even to insert a random sector number on MDV files (I assume that this is to cope with MDV programs which are copy protected with a mechanism which detects a particular random number placed on a MDV tape when it was formatted).

QLAYT can do a surprising number of things and all the possible commands and permutations would probably warrant a complete book, not just an article, to itself.

QLayer 2 has a Windows menu system replacing combination key presses and some command line options in older versions of QLayer. For example, in older versions of QLayer, you could press CTRL ALT SHIFT and R to force a reset, like pressing the reset button on a QL. In this version, there is a menu bar across the top of the emulator display and this includes a reset command.

Back to QLayer2 itself and how it works in practice. Compatibility with existing software is quite good, once you have successfully transferred them to QLayer2 directories and configured everything to run from WIN. What is less good is the sluggishness of the PC when trying to run something else, and how QLayer2's screen update can become quite sluggish or even stop altogether when running intensively, the only way to restore the screen when this happens seems to be to click outside of QLayer and then back into QLayer, and I'm not quite sure why unless it's something to do with suspending QLayer processing and giving I/O a chance to catch up.

Things which don't work at all include sound (no BEEP since the Windows 2000 derivatives of Windows such as XP don't support the required facilities in the kernel architecture), and floppy disc access from within the emulator (you have to use the tools programs). I am not sure about PAR and SER ports, as I couldn't get PAR to work and I have nothing connected to a SER port to test it. The failure to get PAR printing working seemed rather strange as the startup screen lists "NFA file, PAR, SER..." in the ROM identification line at the top of the QL screen (see figure 4) but try as I might to print to "PAR" or "PAR_" I couldn't get it to work.

Things which do work and work well include SuperBASIC and the ability to run existing software. You don't have to cope with the new facilities of SMSQ/E which QPC would force upon you, for example. QLayer 2 cannot run SMSQ/E and is unashamedly good old QDOS, which will probably be important to those who only use the emulator from time to time to run a few favourite programs, or who may be using an emulator for nostalgic reasons - their first "QL" for years. In my case, I only use QLayer2 when I'm away from home - I revert to using the QL at home.

Was it worth the installation? Given that it was free, and the bits which work do work very well indeed, QLayer 2 is a simple and perfectly adequate way to run QDOS on a PC. It has filled my need for a system to let me run QL software on a second and more portable machine in the form of our laptop. I've felt a bit frustrated by the lack of direct floppy disc handling, but once I got used to using the hard disc via the WIN drives, I found I didn't need the floppy discs from day to day, so I only needed to resort to the tools programs every once in a while if I wanted to copy something to carry over to the main QL via floppy disc. It was handy being able to transfer files such as documents between Windows and QDOS so simply, using the QLAYT program. QLayer2 ran most of the QDOS programs I threw at it, including many of those needing pointer environment. It can easily match the speed of a standard QL on most fairly modern PCs, even exceed it in some cases. Usefully, you can also slow it down for playing fast games, a great help to someone of my age of course!

I wouldn't have paid big money for this emulator, but given that it's free I found it to be quite adequate for the things I needed of it - and I would pay it even higher praise if parallel port printing could be made to work and it could read QL floppy disks directly in the emulator.

So, I reckon I'll be keeping in touch with my cute little holiday romance in the future!

And here is how you can share the romance - download QLayer2 from this website:

<http://www.jadnam.org/QL/QLAY2/>

Curve Recognition

by Stephen Poole

In a recent QL Today, bike_bas demonstrated how curve interpolation can be used to reconstruct curves from just a few points scattered around perimeters. This is like a form of image compression, as vector coordinates take up less space than total bitmap lists. To operate, interpolation needs to work out how successive sections of curves lie in relation to one another, and this is the first step toward curve recognition: Type selection. Once you know the relative positioning of sections of a curved line, all you have to do is to compare them to lists of character curve-components, and therefore you are on the road to Optical Character Recognition, (OCR). So I wrote a program that would define curvilinear shapes, and to prove it, one that would fill them no matter how imbricated they were. To do this, I had to set up a graphics-input 'front-end' that would allow me to draw shapes interactively, together with a module that would convert those shapes directly into compressed DATA statements that I could then save to disk, and subsequently reload into the bottom end of the program. Most of this code has been removed from the demo, as it is too lengthy for you to type in, and this explains why the remaining code may seem curiously written: It is left compatible with the original drawing and debugging package!

Now for the secret of how it functions: The program seeks to unravel the interwoven curve by defining it as a clockwise closed trajectory: To do this it has to 'blank out' any 'four-way junction' points, so that the line is continuous from end to end. Dead simple once you have realised it!

So the program first finds all points that need to be deleted, and then indicates the total trajectory, storing pixels by their relative positions in a '3 by 3 square' surrounding the current point:

```
812  
703  
654
```

Once the shape has therefore been defined as a series of closed loops, the inner edges of those loops can be detected and thereby filled correctly by the QL. (I did not want to have to resort to using recursive pixel finding routines which sim-

ply do brute-force searching). The program pauses drawing using 'INKEY\$' so you can see what is going on, (otherwise with QPC, output is virtually instantaneous).

The front-end originally used the magnified shape to help get the drawing right, but vector-fonts must be poked into reserved RAM areas, so the program draws a pixel-compatible reduced shape ready for transfer into memory. After much experimenting, I found that the routines worked satisfactorily, but as I was only programming as a hobby at that time, (and living in perfect isolation in France too), I made no effort to circulate the method, and did not develop it further. OCR was in its infancy at that time, but has now become commonplace on PC's. Experiment as much as you like with this program.

If you find OCR is easy, why not try your hand at Automatic Machine Translation, (MT). This has proved to be far harder than anybody imagined: The best MT programs are at best approximations, sometimes confusing and often downright false! This is why so many manuals for the latest electronic gadgetry are frequently utter rubbish! No wonder Webmasters get few replies from foreign research workers when they use MT! My initial motivation for writing this program was to write and read handwriting on the QL. Although I did not have time to finish the project, I acquired enough understanding of the problems to see how to do it. The exact definitions of the entire alphabet as 'curve-fragments' would have taken too long to enter as DATA statements, otherwise I would have finished the exercise. It would be much simpler to SElect the curve interpolation 'types' rather than to compare chains of strung-together pointers so as to identify arcs. Indeed the main breakthrough was the original construction and definition of curve-interpolation, which took me about a month of evening programming-time to set up... as although the principle is quite straightforward once you have defined it, it takes masses of tight coding to implement. No doubt there are other uses for interpolation just waiting to be realised. If you have any working Vector Font programs, please contact the Editor, as these are badly needed on the QL at present.

```

110 REMark RECOGNITION_bas by S.Poole. v2sep92, v6sep06
120 CLEAR: OPEN#1,con_256
130 WINDOW 256,206,128,0: BORDER 1,2: PAPER 4: CLS
140 iwt=19: i=0: tx$='CAB': wd=4: hg=4
150 FOR fl=0,1: write tx$
160 i$=INKEY$(#1,-1): STOP: REMark QUIT
170 :
180 DEFine PROCedure write(wrt$)
190 LOCAL s$,ls,f,C$
200 REMark write out a string of symbols:
210 s$=wrt$: ls=LEN(s$): oldx=0: oldy=0
220 FOR fff=1 TO ls
230 C$=wrt$(fff): symbol_print C$: i$=INKEY$(#1,150): CLS
240 END FOR fff
250 END DEFine
260 :
270 DEFine PROCedure symbol_print(tx$)
280 LOCAL a,d
290 REMark Put each curved shape in its rectangle:
300 s$=(PICK_(tx$)): get_box_size s$
310 a=ac+1: d=dn+1: xo=a: yo=d
320 DIM g$(a,d),j$(a,d),z$(a,d)
330 IF fl=0 THEN
340 grid: put_strg_in_box s$
350 ELSE
360 grid: put_strg_in_box s$: surface_fill s$
370 END IF
380 END DEFine
390 :
400 DEFine PROCedure surface_fill(sfl$)
410 LOCAL s$: s$=sfl$
420 REMark Find and fill each concave area:
430 get_perimeter s$: edge: solid
440 put_strg_in_box s$
450 END DEFine
460 :
470 DEFine FuNction PICK_(p$)
480 LOCAL p,n,ls,C$
490 REMark Get each shape's vector string:
500 C$='ABCD': p=0+(p$ INSTR C$)
510 RESTORE 1910: READ n: IF p>n OR p=0: RETURN '0'
520 RESTORE 1910+(p*10): READ rd$: ls=LEN(rd$)
530 cpy='*' INSTR rd$: org$=rd$(cpy TO ls): rtn$=rd$(1 TO cpy-1)
540 xA='_x' INSTR org$: xZ='x_' INSTR org$: orgX=org$(xA+2 TO xZ-1)
550 yA='_y' INSTR org$: yZ='y_' INSTR org$: orgY=org$(yA+2 TO yZ-1)
560 RETURN rtn$
570 END DEFine
580 :
590 DEFine PROCedure solid(p1)
600 REMark Fill the inner area of each part:
610 FOR y=1 TO dn
620 x$=' ': lx=ac+1: FOR x=1 TO ac: x$=x$&z$(x,y)
630 REPEAT xrow
640 LF='FO' INSTR x$: IF LF=0: EXIT xrow
650 x=LF+1: x$(LF)=' ': RG='OF' INSTR x$
660 IF RG: x$(RG)=' ': lg=RG-LF: Bk lg,7: Bk lg,8
670 END REPEAT xrow
680 END FOR y
690 END DEFine
700 :
710 DEFine PROCedure get_box_size(stg$)
720 LOCAL x,y,f,s$
730 REMark add vector-types to determine grid-width and height:

```

```

740 s$=stg$: minX=0: maxX=0: minY=0: maxY=0: ls=LEN(s$): x=0: y=0
750 FOR f=1 TO ls
760   ch$=s$(f): IF ch$='9': GO TO 810: ELSE sl=ch$
770   SElect ON sl: =2,3,4: x=x+1: =6,7,8: x=x-1
780   SElect ON sl: =1,2,8: y=y-1: =4,5,6: y=y+1
790   IF x<minX: minX=x: ELSE IF x>maxX: maxX=x
800   IF y<minY: minY=y: ELSE IF y>maxY: maxY=y
810 END FOR f
820 ac=maxX-minX+3: dn=maxY-minY+3: xo=ac+1: yo=dn+1
830 END DEFine
840 :
850 DEFine PROCedure put_strg_in_box(stg$)
860 LOCAL x,y
870 REMark Draw shape from unstrung vectors:
880 x=orgX: y=orgY: ls=LEN(stg$)
890 FOR f=1 TO ls
900   ch$=s$(f): IF ch$='9': GO TO 940: ELSE sl=ch$
910   SElect ON sl: =2,3,4: x=x+1: =6,7,8: x=x-1
920   SElect ON sl: =1,2,8: y=y-1: =4,5,6: y=y+1
930   Bk 7,1: Bk 7,2: g$(x,y)=7: j$(x,y)=7: z$(x,y)=7
940 END FOR f
950 END DEFine
960 :
970 DEFine PROCedure get_perimeter
980 LOCAL x,y,f,loop
990 FOR x=1 TO ac: FOR y=1 TO dn: crossout: END FOR y: END FOR x
1000 x=orgX: y=orgY: t$='0': Bk 0,1: Bk 0,2
1010 REMark procede clockwise from first point:
1020 REPEAT loop
1030   FOR f=1 TO 8
1040     ok=0: j=f
1050     xx=x: SElect ON j=2,3,4: xx=x+1: =6,7,8: xx=x-1
1060     yy=y: SElect ON j=1,2,8: yy=y-1: =4,5,6: yy=y+1
1070     IF z$(xx,yy)='7' THEN
1080       t$=t$&j: x=xx: y=yy: z$(x,y)='i': ok=1
1090       Bk 0,1: Bk 0,2: EXIT f
1100     END IF
1110   END FOR f
1120   IF ok=0: EXIT loop
1130 END REPEAT loop: t$=t$&'0'
1140 END DEFine
1160 :
1170 DEFine PROCedure crossout
1180 REMark Delete central points on X-junctions:
1190 IF z$(x,y)='7' THEN
1200   IF z$(x-1,y)='7' THEN
1210     IF z$(x+1,y)='7' THEN
1220       IF z$(x,y-1)='7' THEN
1230         IF z$(x,y+1)='7' THEN
1240           z$(x,y)='f': Bk 0,1: Bk 0,2 : BEEP 123,45
1250         END IF
1260       END IF
1270     END IF
1280   END IF
1290 END IF
1300 END DEFine
1310 :
1320 DEFine PROCedure edge
1330 LOCAL x,y,f
1340 REMark Follow edges from outside to inside:
1350 x=orgX: y=orgY
1360 FOR f=1 TO LEN(t$)
1370   a=x+1: B=x-1: c=y+1: d=y-1: sl=t$(f)

```


It was only when I sat down to start writing this column that I realised that I had written an article, review or column in every issue of QL Today and in the last two volumes of IQLR too. Over 13 years! I am amazed by that. We are, by the way, the longest lived QL publication and a lot of that can be laid squarely at the feet of Jochen, who has steadfastly prepared and printed each issue, Dilwyn who edited the magazine for the bulk of its life, Bruce Nicholls who has proof read it and now Geoff our current editor.

Don't worry, this is not my 'so long and thanks for all the fish' moment, it is just me getting a little nostalgic about things. But then Nostalgia isn't what it used to be, is it?

We have seen a lot of changes throughout the life of the QL. Changes in the QL and its hardware and changes in the general world of computing. No one in 1983 would have predicted what lay ahead of them. No one could have foreseen the speed of modern processors and memory, the sheer amount of computing power and of storage space that the average PC user now takes for granted and even complains about. There are now fewer black boxes in general use, although I do still get requests for them and the membranes. As you all well know, emulation is the name of the game these days and I must confess I no longer have a functioning QL set up here. This is not because I didn't want to have one but a user talked me into selling my Super Gold Card and Aurora and I needed the space for a 24 channel mixing desk and recording gear but that is another story.

Birmingham

The Birmingham QL show was actually quite fun in spite of the slight air of chaos when we arrived. It seems that the original venue for the show was a largish hall which could be divided in two. It transpired that a wedding party had been booked into the other half and the noise and disruption that would have caused would have been bad for the show. As a result we were booked into a series of small suites but that actually worked out very well. Simon Goodwin managed a marathon presentation and gave a away a whole computer as a prize. It was good to see Rich Mellor there looking healthy. I don't

think any of us sold much but that is what I expect these days. It was a good show socially and I enjoyed most of the day. Well done and a big thank you to the organisers.

Another Quantum Leap

Other items in the news in the weeks before I finished this article included one of those on-going 'pundit-fests' on the Future of Computing. You may wonder why this is of any interest to us. Well it is because it is the long debated 'Quantum Computer Project'. I am repeating here an extract from the BBC website

'A quantum computer is a theoretical device that would make use of the properties of quantum mechanics, the realm of physics that deals with energy and matter at atomic scales.

In a quantum computer data is not processed by electrons passing through transistors, as is the case in today's computers, but by caged atoms known as quantum bits or Qubits.

"It is a new paradigm for computation," said Professor Artur Ekert of the University of Oxford. "It's doing computation differently."

A bit is a simple unit of information that is represented by a "1" or a "0" in a conventional electronic computer.

A qubit can also represent a "1" or a "0" but crucially can be both at the same time - known as a superposition.

This allows a quantum computer to work through many problems and arrive at their solutions simultaneously.

"It is like massively parallel processing but in one piece of hardware," said Professor Ekert.

There are several aspects of this that amuse me. Right at the start they say it is a 'theoretical device' - if only Sir Clive had been that honest when he announced the QL. Then there is the qubit can be a 1 or a 0 or both concept. That's going to be accurate then isn't it? The last part of this is the 'Q' concept. Now why did we not copyright the idea of spelling things with a 'q'. We could make enough money, if this takes off, to finance a whole new QL line - how about a Quantum QL made by Quanta? Mind you we would still be hamstrung by the old problem of not knowing when it will be released. The Uncertainty Principle or Schrodingers Cat ate my mouse.

Wafer thin Mint anyone?

Last issue's article on the ill fated Sinclair wafer technology designs stirred a lot of people's imagination. It was quite by chance that I came across this small news video featuring the announcement of that same project on the BBC's Micro show in 1985. You can watch the item here:

http://www.youtube.com/watch?v=li0I2i_yWMc

or

tinyurl.com/yoqxhd

It is interesting to see what they thought way back then about the future of computing - and to see the hairstyles.

And now as you wait with bated breath I would like to announce my Traditional Christmas punishment:

Christmas Books and Films for Computer buffs: Children's Section

Flushed Away - available at your local Cache and Carry

Transformers - Tony Firshman's vast Sinclair PSU Collection on Film

Harry Potter and the Order from Phoebus - starring Geoff Wicks as Dumbledore and David Gilham as the boy wizard.



In the action and horror section we have:

Hackacoda - Feast of Jarva

8 weeks later - They tried to load a microdrive

Black Sheep - RAM from a suspect source

Bug - The Microsoft Story

Everything's Gone Green - The monitor cable is bust again.

Hot Fuzz - A new pointer command

Superbad - disk corruption like you would not believe

Dances with Wolf(gang Lenerz)

Raid in Entebbe - this will get your backup

Moby Duck - a film about the Chinese Restaurant in Eindhoven.

And here are a couple of books for Christmas:

Geoff Wicks - Secret Diary of A Mole.

Simon Goodwin - Mice Code in Linux

And lastly the QL related CD compilation:

Tony Firshman - 'Careless Whiskers'

John Mason and the Gilpins - 'Quondo, Quanta, Quando'

Dilwyn Jones - I'll Keep a QL in the Hillside

Geoff Wicks - Spies Up Your Life (Spice Girls cover)

Bob Spelten - A DAY(TAbase) in the Life

Jochen Merz - Eat The Menu (Extended) (Bjork cover)

featuring:

Marcel Kilgus - Pizza Carto Violin

Clive Sinclair re-release - Is the the Wafer Amarilo?

Rich Mellor - Re-Membrance the Good Times (Scorpions cover)

With Special Guest appearances by:

Bill Gates - The Windows of your Mind

Linus Torvald - I want to be Free

The Sinclair team sing the press release for the schedule for release of the original QL.

Sinclair research - Until the 1st of Never

The QL Show Agenda

QL Meetings in Eindhoven

Saturday, 22nd of March 2008, 11:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

Thanks to the organiser, Sjef van de Molengraaf, the meetings at Eindhoven also continue in 2008. Same venue as always J-M-S will be there, as always. The dates for the other shows are already fixed, so that you can add it to your agenda.

Saturday, 14th of June 2008, 11:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

Saturday, 18th of October 2008, 11:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

Please note that the 2008 shows now start at 11:00 instead of 10!

**Happy Qling
in 2008!**

The Next Issue

We plan to have the next issue ready for you before the middle of next March. As always, it depends on how quickly we get reviews, articles etc.

We need more material, as always. The more we get and the sooner we get it, the quicker the next issue will be in your hands, and the better it will be. Hope to meet you at one of the forthcoming QL shows - your QL Today Team!